IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	:	THE COMMISSIONER IS AUTHORIZED TO CHARGE ANY DEFICIENCY IN THE FEES FOR THIS PAPER TO DEPOSIT ACCOUNT NO. 23-0975
Kohsaku SHIBATA	:	
Serial No. NEW	:	**Attn: APPLICATION BRANCH**
Filed December 9, 2003	:	Attorney Docket No. 2003_1791A

SIMULATION APPARATUS, METHOD AND PROGRAM

CLAIM OF PRIORITY UNDER 35 USC 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicant in the above-entitled application hereby claims the date of priority under the International Convention of Japanese Patent Application No. 2002-360362, filed December 12, 2002, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Kohsaku SHIBATA

By _____
Michael S. Huppert
Registration No. 40,268
Attorney for Applicant

MSH/kjf
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
December 9, 2003

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００２年１２月１２日

出願番号
Application Number: 特願２００２－３６０３６２

[ST.10/C]: [ＪＰ２００２－３６０３６２]

出願人
Applicant(s): 松下電器産業株式会社

２００３年　５月２０日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３０３７４８８

【書類名】　特許願
【整理番号】　5037740114
【あて先】　特許庁長官殿
【国際特許分類】　G06F
【発明者】
【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
【氏名】　柴田　耕作
【特許出願人】
【識別番号】　000005821
【氏名又は名称】　松下電器産業株式会社
【代理人】
【識別番号】　100109210
【弁理士】
【氏名又は名称】　新居　広守
【電話番号】　06-4806-7530
【手数料の表示】
【予納台帳番号】　049515
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1
【包括委任状番号】　0213583
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　シミュレーション装置、シミュレーション方法及びプログラム

【特許請求の範囲】

【請求項１】　シミュレーション対象のＶＬＩＷプロセッサによって同時実行されるべき複数命令に対して、当該複数命令の逐次実行に対応する命令単位のシミュレーション結果を生成する

ことを特徴とするシミュレーション装置。

【請求項２】　前記シミュレーション装置は、

同時実行されるべき複数命令からなる命令群をシミュレーションする第１シミュレーション手段と、

第１シミュレーション手段によるシミュレーション結果に基づいて、前記命令群中の命令毎のシミュレーション結果を生成する第２シミュレーション手段と

を備えることを特徴とする請求項１記載のシミュレーション装置。

【請求項３】　前記第１シミュレーション手段は、

ＶＬＩＷプロセッサのリソースデータを示す第１データを記憶する記憶手段と

第１データのコピーを第２データとして記憶手段に保存する保存手段と、

保存後に、１つの命令群の実行をシミュレーションすることにより第１データを更新する第１シミュレータと

を備え、

前記第２シミュレーション手段は、第１データおよび第２データに基づいて、当該命令群中の命令について命令単位のシミュレーション結果を得る

ことを特徴とする請求項２記載のシミュレーション装置。

【請求項４】　前記保存手段は、レジスタセットのデータを第２データとして記憶手段に保存し、

前記第２シミュレーション手段は、前記命令群中の命令単位に当該命令のシミュレーション実行前の状態にリソースを復元する

ことを特徴とする請求項３記載のシミュレーション装置。

【請求項５】　前記保存手段は、さらに、前記命令群にメモリライト命令が存在する場合、ライト前のメモリデータを第２データに含ませて記憶手段に保存する

ことを特徴とする請求項４記載のシミュレーション装置。

【請求項６】　前記第２シミュレーション手段は、

ブレーク条件を満たす命令が第１シミュレーション手段によりシミュレーションされた命令群に含まれているか否か判定する判定手段と、

含まれていないと判定された場合に、第１シミュレーション手段に次の命令群のシミュレーションを指示する指示手段と、

含まれていると判定された場合に、前記ブレーク条件を満たす命令を停止命令と決定する決定手段と

を備えることを特徴とする請求項５記載のシミュレーション装置。

【請求項７】　前記判定手段は、命令単位のシミュレーションのステップ実行において、現在の停止命令の次の命令をブレーク条件とする

ことを特徴とする請求項６記載のシミュレーション装置。

【請求項８】　前記第２シミュレーション手段は、さらに、

前記第１データ及び第２データに基づいて、決定手段に決定された停止命令の直前の命令までシミュレーションした場合のリソースデータを復元する復元手段を備える

ことを特徴とする請求項６記載のシミュレーション装置。

【請求項９】　前記第２シミュレーション手段は、さらに、

前記第１データ及び第２データに基づいて、決定手段に決定された停止命令までシミュレーションした場合のリソースデータを復元する復元手段を備える

ことを特徴とする請求項６記載のシミュレーション装置。

【請求項１０】　前記第１シミュレータは、前記命令群中の各命令について当該命令により変更されるリソースを示す更新情報を生成し、

前記復元手段は、第１データ、第２データ及び更新情報に従って当該命令群中の命令毎に当該命令までの逐次実行した結果に対応するリソースデータを復元する

ことを特徴とする請求項８又は９記載のシミュレーション装置。

【請求項１１】　前記第１シミュレータは、パイプライン処理を行うＶＬＩＷプロセッサを対象として、パイプライン処理のサイクル単位で命令群のシミュレーションを行い、

前記シミュレーション装置は、さらに、命令群単位のシミュレーションにおける実行サイクル数をカウントする

ことを特徴とする請求項３記載のシミュレーション装置。

【請求項１２】　前記ＶＬＩＷプロセッサは、同時実行されるべき複数命令中の命令実行をキャンセルするキャンセル機構を有し、

前記第１シミュレータは、キャンセル機構をシミュレーションする

ことを特徴とする請求項１１記載のシミュレーション装置。

【請求項１３】　前記第１シミュレータは、さらに、シミュレーション対象のＶＬＩＷプロセッサの実行ステージで遅延サイクルを発生させる遅延命令ついて、遅延サイクルをシミュレーションし、

前記復元手段は、遅延命令についての更新情報に従って遅延命令の実行結果に対応するリソースデータを生成する

ことを特徴とする請求項１１記載のシミュレーション装置。

【請求項１４】　前記復元手段は、さらに、シミュレーション対象のＶＬＩＷプロセッサの実行ステージで遅延サイクルを発生させる遅延命令との間で、同一命令群内で出力依存関係を有する出力依存命令について、遅延命令についての更新情報と出力依存命令についての更新情報に従って、出力依存命令の実行結果に対応するリソースデータを生成する

ことを特徴とする請求項１３記載のシミュレーション装置。

【請求項１５】　シミュレーション対象のＶＬＩＷプロセッサによって同時実行されるべき複数命令に対して、当該複数命令の逐次実行に対応する命令単位のシミュレーション結果を生成する

ことを特徴とするシミュレーション方法。

【請求項１６】　前記シミュレーション方法は、

同時実行されるべき複数命令からなる命令群をシミュレーションする第１ステ

ップと、

第１ステップにおけるシミュレーション結果に基づいて、前記命令群中の命令毎のシミュレーション結果を生成する第２ステップと

を有することを特徴とする請求項１５記載のシミュレーション方法。

【請求項１７】　前記第１ステップは、

ＶＬＩＷプロセッサの現在のリソース内容を第１データのコピーを第２データとして記憶部に保存する保存サブステップと、

保存後に、前記命令群の実行をシミュレーションすることにより第１データを更新する更新ステップと

を有し、

前記第２ステップにおいて、第１データおよび第２データに基づいて、前記命令群中の命令単位のシミュレーション結果を得る

ことを特徴とする請求項１６記載のシミュレーション方法。

【請求項１８】　前記保存サブステップにおいて、レジスタセットのデータを第２データとして記憶部に保存し、

前記第２ステップにおいて、前記命令群中の命令単位に当該命令のシミュレーション実行前の状態にリソースを復元する

ことを特徴とする請求項１７記載のシミュレーション方法。

【請求項１９】　前記第２ステップにおいて、

ブレーク条件を満たす命令が第１ステップにおいてシミュレーションされた命令群に含まれているか否か判定する判定サブステップと、

含まれていないと判定された場合に、第１ステップにおいて次の命令群のシミュレーションを再度行わせ、含まれていると判定された場合に、前記ブレーク条件を満たす命令を停止命令と決定する決定サブステップと

を有することを特徴とする請求項１８記載のシミュレーション方法。

【請求項２０】　前記更新ステップは、パイプライン処理を行うＶＬＩＷプロセッサを対象として、パイプライン処理のサイクル単位で命令群のシミュレーションを行い、

前記シミュレーション方法は、さらに、命令群単位のシミュレーションにおけ

る実行サイクル数をカウントする

ことを特徴とする請求項１９記載のシミュレーション方法。

【請求項２１】　請求項１５ないし２０の何れか１項に記載のシミュレーション方法をコンピュータに実行させることを特徴とするプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

ＶＬＩＷ（Very Long Instruction Word）プロセッサ用のプログラムをシミュレーションすることによりソフトウェア開発者のプログラム開発を支援するシミュレーション装置に関する。

【０００２】

【従来の技術】

プロセッサの実行状態を模倣するシミュレーション装置は、プログラム開発において有用である。パイプライン制御を行うプロセッサを対象とするシミュレーション装置では、パイプラインのシミュレーションを正確に行うため、サイクル単位でのシミュレーションを行っている。これに関する先行技術文献として特許文献１がある。

【０００３】

この特許文献１に開示されたパイプライン計算機シミュレータは、パイプラインのシミュレーションを行いながら１命令単位でのステップ動作のシミュレーションをおこなう。これにより、デバッグ動作に有用な１命令のステップ動作が直感的に正しく動作する。

【０００４】

【特許文献１】

特開平８－２７２６１２

【０００５】

【発明が解決しようとする課題】

しかしながら、上記従来技術では、単一命令発行のプロセッサを対象として命令単位のステップ動作を行うものの、複数命令を同時に実行するプロセッサを対

象とするシミュレーション装置は、複数命令の同時実行をシミュレーションすることから、ソフトウェア開発者は命令単位の実行過程まで細かく確認することができなかった。

【０００６】

本発明は、複数命令を同時実行するプロセッサを対象とし、命令単位の実行過程がわかるようにシミュレーションを行うシミュレーション装置を提供することを目的とする。

【０００７】

【課題を解決するための手段】

上記課題を解決するため本発明のシミュレーション装置は、シミュレーション対象のＶＬＩＷプロセッサによって同時実行されるべき複数命令に対して、当該複数命令の逐次実行に対応する命令単位のシミュレーション結果を生成するよう構成されている。

【０００８】

この構成によれば、複数命令を同時実行するプロセッサにおける命令実行動作を、命令単位の実行過程まできめ細かくデバッグ及び検証するのに役立つ。例えば、同時に実行される命令を個別に１命令ずつデバッグすることができる。

【０００９】

また、前記シミュレーション装置は、同時実行されるべき複数命令からなる命令群をシミュレーションする第１シミュレーション手段と、第１シミュレーション手段によるシミュレーション結果に基づいて、前記命令群中の命令毎のシミュレーション結果を生成する第２シミュレーション手段とを備える構成としてもよい。

【００１０】

この構成によれば、命令群のシミュレーション結果を利用して、命令群中の個々の命令についてシミュレーション結果を生成可能であり、ソフトウェア開発者に対しては、あたかも命令単位に逐次シミュレーションしているように見せることができるので、命令個別のデバッグ及び検証を容易にすることができる。

【００１１】

さらに、前記第１シミュレーション手段は、ＶＬＩＷプロセッサの現在のリソースを第１データとして記憶する記憶部と、第１データのコピーを第２データとして記憶手段に保存する保存手段と、保存後に、前記命令群の実行をシミュレーションすることにより第１データを更新する第１シミュレータとを備え、前記第２シミュレーション手段は、第１データおよび第２データに基づいて、前記命令群中の命令単位のシミュレーションを結果を得る構成としてもよい。

【００１２】

この構成によれば、１つの命令群のシミュレーション前のリソースデータを示す第２データとシミュレーション後のリソースデータを示す第１データとに基づいて、命令単位のシミュレーションを結果を得ている。

【００１３】

また、前記第２シミュレーション手段は、ブレーク条件を満たす命令が第１シミュレーション手段によりシミュレーションされた命令群に含まれているか否か判定する判定手段と、含まれていないと判定された場合に、第１シミュレーション手段に次の命令群のシミュレーションを指示する指示手段と、含まれていると判定された場合に、前記ブレーク条件を満たす命令を停止命令と決定する決定手段とを備える構成としてもよい。

【００１４】

この構成によれば、同時実行される複数命令に対して命令個別にブレークさせることが可能なので、ソフトウェア開発者にとって、同時実行される複数命令における命令の相互関係を容易に確認することができる。また、シミュレーション済の命令群に含まれる命令を対象にブレーク条件を判定しているので、プロセッサにおける実際の実行経路においてブレークさせることができる。例えば、命令群中にキャンセルされた命令がある場合には、実行されない命令を誤ってブレークさせることがない。

【００１５】

ここで、前記第１シミュレータは、パイプライン処理を行うＶＬＩＷプロセッサを対象として、パイプライン処理のサイクル単位で命令群のシミュレーションを行い、前記シミュレーション装置は、さらに、命令群単位のシミュレーション

における実行サイクル数をカウントする構成としてもよい。

【００１６】

この構成によれば、パイプライン処理するプロセッサを対象に、命令単位のシミュレーション結果を生成しながらも、プロセッサのパイプラインサイクル数を正確にシミュレーションすることができる。

【００１７】

また、前記第１シミュレータは、前記命令群中の各命令について当該命令により変更されるリソースを示す更新情報を生成し、前記復元手段は、第１データ、第２データ及び更新情報に従って当該命令群中の命令毎に当該命令までの逐次実行した結果に対応するリソースデータを復元する構成としてもよい。

【００１８】

この構成によれば、更新情報を利用することによりリソースデータの復元を容易にすることができる。

ここで、前記第１シミュレータは、さらに、ＶＬＩＷプロセッサの実行ステージで遅延サイクルを発生させる遅延命令ついて、遅延サイクルをシミュレーションし、前記復元手段は、遅延命令についての更新情報に従って遅延命令の実行結果に対応するリソースデータを生成する構成としてもよい。

【００１９】

この構成によれば、プロセッサにおいては、同時実行される複数命令中の遅延命令は遅れて実行されるので、前記複数命令はサイクルレベルでは同時ではなく実行順序が入れ替わるが、命令単位のシミュレーションにおいては１つずつ順に実行したシミュレーション結果を生成することができる。

【００２０】

さらに、前記復元手段は、さらに、シミュレーション対象のＶＬＩＷプロセッサの実行ステージで遅延サイクルを発生させる遅延命令との間で、同一命令群内で出力依存関係を有する出力依存命令について、遅延命令についての更新情報と出力依存命令についての更新情報に従って、出力依存命令の実行結果に対応するリソースデータを生成する構成としてもよい。

【００２１】

この構成によれば、プロセッサにおいては、同時実行される出力依存関係にある命令の一方の実行結果はキャンセルしたのと同じ結果になるが、命令単位のシミュレーションでは、出力依存関係によりキャンセルされる運命にある命令についてもシミュレーション結果を生成するので、ソフトウェア開発者は、結果的に出力依存命令が結果的にキャンセルされる過程を知ることができる。

【００２２】

【発明の実施の形態】

＜シミュレーションシステムの概要＞

図１は、本発明の実施の形態におけるシミュレーションシステム１の外観を示す図である。同図のシミュレーションシステム１は、シミュレーション装置２とデバッグ装置３とからなる。

【００２３】

シミュレーション装置２は、本体装置２ａ、表示装置２ｂ、入力装置２ｃを備え、複数命令を同時に実行するプロセッサの動作をシミュレーションする装置であって、プロセッサでは同時実行される複数命令について、命令を一つずつ順に実行した場合に得られる実行結果をＬＡＮケーブルを介してデバッグ装置３に通知するよう構成されている。すなわちシミュレーション装置２は、ユーザからデバッグ装置３を介して命令単位のステップ動作指示や、命令単位のブレークポイント指示を受け付けて、命令単位の実行結果をデバッグ装置３を介してユーザに対して表示する。本体装置２ａは、シミュレーションソフトウェアを実行することにより上記のシミュレーションを行う。表示装置２ｂ、入力装置２ｃはシミュレーションソフトウェアの実行開始と終了に用いられる。

【００２４】

デバッグ装置３は、本体装置３ａ、表示装置３ｂ、入力装置３ｃを備え、シミュレーション装置２に対するユーザインターフェースとして機能し、ユーザから、命令単位のステップ動作や、命令単位のブレークポイント指定などの操作を受けてシミュレーション装置２に通知し、シミュレーション装置２からの命令単位のシミュレーション結果を受信して表示する。本体装置３ａは、デバッグソフトウェアを実行することにより、ユーザインターフェース機能を提供し、シミュレ

ーション装置２に対してシミュレーション実行指示やその応答を送受信する。表示装置３ｂは、命令単位のシミュレーション実行結果などを表示する。入力装置３ｃは各種ユーザ操作を受け付ける。

【００２５】

図２はシミュレーション装置２においてシミュレーションソフトウェア起動後の表示内容の一例を示す。同図において、Ｗ０はコンソールウィンドウであり、シミュレーションソフトウェアが起動されて実行中の状態を示している。この状態において、デバッグ装置３を介してソフトウェア開発者が本シミュレーション装置２を利用可能となる。

【００２６】

図３は、デバッグ装置３においてデバッグソフトウェア実行中の表示内容の一例を示す。同図において、Ｗ１はデバッグ対象のプログラムを表示するコード表示ウィンドウ、Ｗ２はデバッグ対象のソースプログラムを表示するソース表示ウィンドウ、Ｗ３はユーザ操作により各種シミュレーション用のコマンドを入力するためのコマンド入力ウィンドウ、Ｗ４はターゲットプロセッサにおいて実行した場合のサイクル数とシミュレーションした命令数を示すステップ数等を表示する状態表示ウィンドウ、Ｗ５はレジスタデータを表示するレジスタ内容表示ウィンドウ、Ｗ６はメモリデータを表示するメモリ内容表示ウィンドウである。Ｍ１は命令単位のシミュレーションにおけるシミュレーション未実行の先頭命令（以下、停止命令と呼ぶ）を示す停止命令マークであり、Ｍ２は停止命令に対応するソースプログラム中のプログラム行を示す停止行マークである。

【００２７】

コード表示ウィンドウＷ１は、デバッグ対象のプログラムについて命令アドレスを示すプログラムカウント値（同図のＰＣ欄）、行番号（ＬＩＮＥ欄）、フラグ（ＦＬＧ欄）、ニーモニック（ＭＮＥＭＯＮＩＣ欄）等を表示するとともに、停止命令マークＭ１を表示している。ここで、フラグ［Ｆ０］は、ＶＬＩＷプロセッサ２００に備えられた状態フラグレジスタ中のフラグＦ０の値により命令が有効か否かを示し、直前の比較命令（ｃｍｐ命令）における比較結果に応じてリセットされ得る。このフラグＦ０は、条件分岐命令における条件の成否をフラグ

Ｆ０の値に反映させることにより、例えばソース表示ウィンドウＷ２中のｉｆ文の条件に依存する実行文などに利用される。またコード表示ウィンドウＷ１における；；（セミコロン２つ）は、ターゲットプロセッサにおいて同時実行される命令の切れ目を示し、例えばデータ依存関係のある２つの命令など同時実行できない直後の命令を次の命令群に分けるために使用される。

【００２８】

なお、図１に示したシミュレーションシステム１は、シミュレーションソフトウェアを実行するコンピュータ（シミュレーション装置２）とデバッグソフトウェアを実行するコンピュータ（デバッガ３）の２台のコンピュータから構成されるが、１台のコンピュータによりシミュレーションソフトウェアとデバッグソフトウェアの両方を実行する構成としてもよい。

＜ターゲットプロセッサ＞

次に、シミュレーションシステム１のターゲットプロセッサの仕様について図４～図１１を用いて説明する。

【００２９】

図４は、シミュレーション対象のＶＬＩＷプロセッサの構成を示すブロック図である。同図のように、プロセッサ２００は、ＩＦ（命令フェッチ）ステージ、ＤＣ（デコード）ステージ、ＥＸ／ＭＥＭ（実行／メモリアクセス）ステージ、ＷＢ（ライトバック）ステージからなる４段パイプライン構成であり、ＩＦステージにおいて複数命令からなる命令群をフェッチする命令フェッチ制御部２０１と、ＤＣステージにおいてフェッチされた最大３個の命令を同時にデコードする命令デコーダ２０２～２０４と、ＥＸステージにおいてデコード結果に従って最大３個の命令を同時に実行するＡＬＵ（Arithmetic Logic Unit）２０５～２０７と、デコードされた命令がメモリアクセス命令である場合にＭＥＭステージにおいてその命令を実行するメモリアクセス制御部２０９と、命令の実行内容あるいは一部の実行内容をキャンセルする命令キャンセル部２２０と、次段のステージへ命令に関する情報を受け渡すパイプラインレジスタ２２１～２２３と、デコードされた命令がメモリアクセス命令である場合にそのデコード情報を選択するマルチプレクサ２３０と、デコードされた命令が分岐命令である場合にそのデコ

ード情報を選択する２３１と、複数の汎用レジスタからなるレジスタファイル２５１と、プログラム及びデータを記憶するメモリ２５２と、フォワーディングユニット２５３とから構成される。

【００３０】

ここで、命令フェッチ制御部２０１は、命令デコーダ２０２～２０４に対して命令アドレスの若い順に命令を配置するよう命令を発行する。これは、命令キャンセル部２２０と相俟って、同時実行される複数命令について、命令デコーダ２０２～２０４の順に命令を実行した場合と同じ結果を得るように構成されているからである。

【００３１】

図５は、ＶＬＩＷプロセッサ２００のパイプライン処理の流れを示す図である。同図において、縦軸（下向き）は実行すべきプログラム中の配置された命令の配置順を、横軸はサイクル数を示す。図中、ＤＣ１～ＤＣ３は命令デコーダ２０２～２０４に、ＥＸ１～ＥＸ３はＡＬＵ２０５～２０７に、ＭＥＭはメモリアクセス制御部２０９において処理されることを示す。ｗａｉｔは、インターロックの発生を意味する。Ｔ１、Ｔ２・・はステージ時間を単位とするサイクルである。また、命令デコーダ２０２及びＡＬＵ２０５を経て命令が処理される経路（パイプラインステージではＩＦ１→ＤＣ１→ＥＸ１→ＷＢ１となる）をスロットｘと呼ぶ。同様にＩＦ２→ＤＣ２→ＥＸ２→ＷＢ２の経路をスロットｙ、ＩＦ３→ＤＣ３→ＥＸ３→ＷＢ３の経路をスロットｚと呼ぶ。スロットｘ中の命令を命令ｘと呼ぶ。スロットｙ、ｚについても同様に命令ｙ、ｚと呼ぶ。

【００３２】

図６は、図５に示した命令１～５の一例を示す。同図では各命令について、命令のアドレス、ニーモニック表記、命令の実行内容、命令実行により更新されるリソースとを記載してある。ここでリソースはレジスタやメモリを含む。同図では関連するリソースのみを示してある。ニーモニック表記のｓｕｂ、ａｄｄ、ｌｄ、ｓｔ、ｏｒは、それぞれ減算命令、加算命令、メモリからレジスタにデータを書き込むロード命令、レジスタデータをメモリに格納するストア命令、論理和命令を意味する。Ｒ０～Ｒ６はレジスタ０～６を、（Ｒ４＋）の＋（プラス）は

実行後に４を加算するポストインクリメントを意味する。また、命令の実行内容の表記において、＝は代入を、ｍｅｍ（Ｒ４）はＲ４の内容をアドレスとするメモリのデータを、｜は論理和をそれぞれ意味する。

【００３３】

また、ストア命令のＭＥＭステージは２サイクルかかり、ロード命令のＭＥＭステージ及び他の命令のＥＸステージは１サイクルで終えるものとする。以下では、ストア命令のようにパイプライン段数（ＶＬＩＷプロセッサ２００では４つ）よりも多くのサイクルを要する命令を遅延命令と呼び、パイプライン段数と同数のサイクルで処理される命令を通常命令と呼ぶ。

【００３４】

なお、説明を簡単にするためストア命令のＭＥＭステージは２サイクル、ロード命令のＭＥＭステージは１サイクルで終わるものとするが、ロード命令及びストア命令のＭＥＭステージのサイクル数が動的に変化する場合でも、本発明においてなんら問題ない。すなわち、ターゲットプロセッサでは、ＭＥＭステージにおいてアクセス先（メモリ素子やｉ／ｏ）からの応答（ACK）をサイクル毎に待っ、応答があったサイクルでメモリアクセスを終えるようにしている。

【００３５】

命令群１は、通常命令である命令１～３からなり、図５に示すようにＴ１～Ｔ４の４サイクルで処理される。命令群２は、通常命令である命令５と遅延命令である命令４を含むので、インターロックの発生により５サイクルの処理時間を要する。

【００３６】

図７は、図６の命令列についてのサイクルと更新されるリソースと関係を示す図である。サイクルＮ＋１では、命令群１の命令１～３が４サイクルで完了するので、命令１～３によりデスティネーションとして指定されたレジスタＲ０、Ｒ２～Ｒ４が更新される。サイクルＮ＋２では、インターロックにより命令群２がまだ完了していないので、更新されるリソースが存在しない。サイクルＮ＋３では、命令群２が完了するので、命令４、５によりデスティネーションとして指定されたｍｅｍ（Ｒ４）、Ｒ４、Ｒ５が更新される。

【００３７】

図８は、命令キャンセル部２２０により命令がキャンセルされる場合の命令列の例を示す図である。同図において命令６、７は同時実行される命令であり、命令８は単独実行される命令である。命令６の比較命令（ｃｍｐ命令）は、フラグＦ０を比較結果に応じてリセット（無効化）する。命令７の加算命令（ａｄｄ命令）は、フラグＦ０が１であれば実行されるが、フラグＦ０が０であれば実行されない。

【００３８】

命令キャンセル部２２０は、同時実行している命令６、７について、命令６の結果に応じて、命令７をキャンセルする。つまり、命令キャンセル部２２０は、命令６によってフラグＦ０が０にリセットされた場合には、命令７の実行結果をレジスタ又はメモリへ書き込みを禁止することにより、その命令７をキャンセルする。

【００３９】

言い換えれば、命令キャンセル部２２０は、同時実行している複数命令中に、条件分岐命令がある場合に条件の成否に応じて条件分岐命令に後続する命令をキャンセルすることできる。こうして、キャンセル部２２０は、同時実行される複数命令中の任意の命令から条件分岐することを可能にしている。例えば、図３に示した停止行マークのある条件文（ｉｆ文）は、停止命令マークのあるｃｍｐ命令とフラグＦ０付きのｍｏｖ命令として実行される（この２命令は命令６、７と同じ使い方である）。

【００４０】

上記の命令６、７は同時に実行される命令であることから本来は先後関係を有していないが、命令キャンセル部２２０は、論理的に命令６の実行が先行しているものとして扱っている。

【００４１】

また、命令キャンセル部２２０は、複数命令中に条件のない単純分岐命令が存在し、分岐により実行されることのない命令が同時に発行された場合でも、その命令をキャンセルする。

【００４２】

命令キャンセル部２２０を備えることによりＶＬＩＷプロセッサ２００は、複数命令を同時実行するプロセッサでありながら、同時実行される複数命令中の任意の命令からの条件分岐可能にするため、同時実行される複数命令の間の関係において命令アドレスの順に命令を実行した場合と同じ結果を得るというアーキテクチャとなっている。

【００４３】

図９は、命令キャンセル部２２０により命令がキャンセルされる場合の命令列について他の例を示す図である。同図において、命令１２～１４は同時実行される命令である。ところが命令１３のロード命令（ｌｄ命令）も命令１４の転送命令（ｍｏｖ命令）もレジスタＲ１をデスティネーションとして指定しており、両命令によるレジスタＲ１への書き込みが競合する関係にある（この関係を出力依存と呼ぶ）。出力依存関係を検出した場合に、命令キャンセル部２２０は、命令アドレスが若い方の命令１３によるレジスタＲ１への書き込みをキャンセルする。その結果、レジスタＲ１には命令１４の実行結果が反映されることになる。この実行結果は、ＶＬＩＷプロセッサ２００が、同時実行する複数命令の間の関係において命令アドレスの順に命令を実行した場合と同じ結果を得るというアーキテクチャを採用していることによる。

【００４４】

図１０は、フォワーディングユニット２５３によるフォワーディングを伴うパイプライン処理を示す説明図である。同図において、ロード命令(ｌｄ　Ｒ２，（Ｒ３＋））及び加算命令（ａｄｄ　Ｒ４，８)は同時に発行され、分岐命令(ｂｒ　Ｒ）は単独で発行されるものとする。また、ロード命令はＭＥＭステージが２サイクルを要するものとする。

【００４５】

加算命令（ａｄｄ　Ｒ４，８)の実行結果Ｒ４が、分岐命令(ｂｒ　Ｒ４）により使用されるので、この２つの命令はデータ依存関係を有している。もし、加算命令のＷＢステージの完了を待って分岐命令を開始するとすれば、データ依存関係に起因して２サイクルのペナルティが発生することになる（ＤＣ１ステージが

Ｔ６のサイクルになる）。

【００４６】

これを回避するため、フォワーディングユニット２５３は、現在実行中の命令群中の何れかの命令（被依存命令と呼ぶ）と次の命令群中の何れか命令（依存命令と呼ぶ）との間にデータ依存関係がある場合に、被依存命令のＥＸステージにおける実行結果として得られたデータを取り込んで一旦保持し、依存命令のＥＸステージ又はＭＥＭステージの開始時に当該データをオペランドデータとして直接出力する（これをフォワーディングと呼ぶ）。

【００４７】

さらに、ＷＢステージで書かれた内容を同じサイクルのＤＣステージで読み出せる構成（Read After Write）にすることで、ＷＢステージの完了を待つことなくデータ依存が存在する後続命令をペナルティなしで実行することができる。

【００４８】

図１０の場合、被依存命令（ａｄｄ　Ｒ４，８)のＥＸ２ステージにおける実行結果として得られたＲ４のデータ”８”は、フォワーディングユニット２５３によってＤＣ１ステージにおいて依存命令(ｂｒ　Ｒ４）のオペランドで指定されたアドレス（８）としてマルチプレクサ２３１を介して命令フェッチ制御部２０１に出力される。ここで、ＤＣ１ステージにおいてデータが出力されているのは、図４に示したようにデコード情報（ＤＣ情報）がＤＣステージの途中からＩＦステージに出力される構成だからである。このように、フォワーディングユニット２５３を備えることによりＶＬＩＷプロセッサ２００では、データ依存関係に起因するペナルティを解消している。

【００４９】

また、図１０において、ロード命令（ｌｄ　Ｒ２，（Ｒ３＋））のＭＥＭステージが２サイクルであることからサイクルＴ４においてメモリアクセスが完了する。また加算命令（ａｄｄ　Ｒ４，８）はＴ３において加算が完了する。この点で、両命令は命令アドレスの順とは異なる順で実行されたことになる。もしＥＸ３ステージをサイクルＴ４で実行したとすれば、当該２命令を命令アドレス順に実行することにはなるが、依存関係がある場合には依存命令(ｂｒ　Ｒ４)にペナ

ルティが生じることになり、遅延することなる。そのため、ＶＬＩＷプロセッサ２００では例外的に命令アドレスの順とは異なる順で実行することがある。その場合でも、実行結果は命令アドレス順に実行した場合と同じ結果を得ることになる。

【００５０】

なお、ターゲットプロセッサは図４に限るものではなく、複数の命令を同時に実行するものであればよい。例えば、図１１に示すＶＬＩＷプロセッサでもよい。図１１のＶＬＩＷプロセッサは、図４に示したＶＬＩＷプロセッサから命令キャンセル部２２０を削除した構成となっている。

【００５１】

＜シミュレーションシステム１の構成＞

図４～図１１に示したターゲットプロセッサを前提に、本発明の実施の形態におけるシミュレーションシステム１についての説明を続ける。

【００５２】

図１２は、シミュレーションシステム１の構成を示す機能ブロック図である。同図においてシミュレーションシステム１は、ユーザインターフェース４、デバッガ３ａ、シミュレーション装置２から構成される。

【００５３】

ユーザインターフェース４は、図１に示した表示装置３ｂ及び入力装置３ｃに相当し、ユーザから命令単位のシミュレーション実行や命令単位のステップ実行を指示する操作を受け付け、そのシミュレーション結果を表示する。もちろん、ユーザインターフェース４は、命令単位ではなく同時実行される複数命令を同時実行するサイクル単位のシミュレーション実行の操作も受け付けた場合は、そのシミュレーション結果を表示する。

【００５４】

デバッガ３ａは、図１に示した本体装置３ａに相当し、ユーザインターフェース４を介して命令単位のステップ実行の指示やブレークポイントの指定などの各種制御コマンド１２８を受け、各種制御コマンドに対する応答としてユーザインターフェース４に介して命令単位のシミュレーション結果を反映したレジスタフ

ァイルの内容１２９やメモリの内容１３０を表示装置３ｂを介して表示する。

【００５５】

また、デバッガ３ａは、シミュレーション装置２に対して命令単位のシミュレーション実行指示１３１や、メモリアドレス及びサイズ１３４をシミュレーション装置２に送信し、シミュレーション実行指示１３１やメモリアドレス及びサイズ１３４に対する応答としてシミュレーション装置２から停止命令通知１３２、レジスタデータ１３３、メモリ内容１３５を受信する。

【００５６】

シミュレーション装置２は、図１に示した本体装置２ａに相当し、図４に示したターゲットプロセッサの動作をサイクル単位で複数命令（命令群）の同時実行をシミュレーションするパイプラインシミュレーション部１０と、そのシミュレーション結果に基づいて命令群中の命令単位のシミュレーション結果を生成する命令シミュレーション部３０とを備える。パイプラインシミュレーション部１０による命令群単位のシミュレーション結果から命令シミュレーション部３０において命令毎の実行前後の状態を生成するという二段階のシミュレーションにより、シミュレーション装置２は、デバッガ３ａに対して、あたかも命令単位にシミュレーションしているかのように命令単位のシミュレーション結果を生成する。

【００５７】

パイプラインシミュレーション部１０は、第１レジスタファイルモジュール１１と、メモリモジュール１２と、共通情報格納部１３と、フェッチモジュール１４と、フェッチ情報格納部１５と、デコードモジュール１６と、デコード情報格納部１７と、実行モジュール１８と、実行情報格納部１９と、完了処理モジュール２０と、完了情報格納部２１と、過去状態更新制御部２２と、スケジューリングモジュール２３とを備える。

【００５８】

命令シミュレーション部３０は、第２レジスタファイルモジュール３１、メモリ値退避部３２、リソース情報変更部３３、シミュレーション制御部３４から構成される。

【００５９】

まず、パイプラインシミュレーション部１０及び命令シミュレーション部３０の各構成要素を説明するのに先立って、同図における各矢線の意味ついて説明する。

【００６０】

１０１はシミュレーション制御部３４からスケジューリングモジュール２３へ出力される１サイクル分の命令群のシミュレーション実行指示、１０２はそのシミュレーション実行指示に対する応答である。１０３は命令群のシミュレーション実行前の状態の一部として第１レジスタファイル１１から第２レジスタファイルモジュール３１へコピーされるレジスタデータである。

【００６１】

１０４はメモリアドレス及びサイズ、１０５は１０４にアドレス指定されたメモリデータであって、もし１０４がストア命令のストア先アドレスであれば当該ストア命令によるストア前のメモリ内容である。１０６はメモリモジュール１２に供給されるメモリアドレス及びサイズ、１０７は１０６にアドレス指定されたメモリデータである。１０８、１０９は１０４、１０５と同内容のデータであり、実行モジュール１８からスケジューリングモジュール２３に供給される。１１０は第２レジスタファイルモジュールのレジスタファイルの内容であり、１１１は第１レジスタファイル１１の内容である。

【００６２】

１１２は実行モジュール１８からの１サイクルのＥＸステージのシミュレーションを実行したことを通知する命令実行通知である。１１３はレジスタ番号及びＲ／Ｗの区別、１１４は１１３によって指定されたレジスタデータである。１１５は第２レジスタファイルモジュール３１を更新してもよいか否かを問う問い合わせ、１１６は第２レジスタファイルモジュール３１の更新を禁止するか否かを示す更新禁止通知である。１１７はメモリモジュール１２に対するアドレス、サイズ、Ｒ／Ｗの区別であり、１１８はメモリ内容、１１７により指定されたメモリデータである。１１９はフェッチ情報、１２０はデコード情報、１２１は実行情報、１２２は完了情報である。

【００６３】

１２３～１２６は、それぞれ完了処理モジュール、実行モジュール、デコードモジュール、フェッチモジュールに対して出力される実行指示であって、この順に出力される（正確には各モジュールが呼び出される）。１２７は、各モジュールで共通に使用される共通情報に含まれるインターロックフラグの値を示す。インターロックフラグはインターロックの発生を意味する。共通情報には、インターロックフラグの他に、パイプラインストールの発生を意味するストールフラグや、分岐命令で指定された分岐先アドレスなどを含む。１２８は各種制御コマンド、１２９は表示用のレジスタデータ、１３０は表示用のメモリデータである。１３１は命令単位のシミュレーション実行指示又はステップ実行指示であり、１３２は１３１への応答として通知される停止命令通知であり、１３３は当該停止命令の実行前の状態を示すレジスタデータ、１３４はメモリアドレス及びサイズ、１３５は１３４によってアドレス指定されたメモリ内容であって当該停止命令の実行前の状態を示す。

【００６４】

続いて、パイプラインシミュレーション部１０及び命令シミュレーション部３０の各構成要素について説明する。

【００６５】

＜パイプラインシミュレーション部１０＞

第１レジスタファイルモジュール１１は、ターゲットプロセッサのレジスタファイルと同じレジスタ構成を有する。

【００６６】

メモリモジュール１２は、ターゲットプロセッサのメモリ構成を有し、デバッグ対象のプログラムを格納する。

共通情報格納部１３は、インターロックが発生したことを示すインターロックフラグや、特定の命令のデータ依存に起因するパイプラインストールの発生を示すストールフラグ等からなる共通情報を格納する。インターロックフラグは、インターロックを発生させたモジュールによってセット／リセットされ、各モジュールから参照される。具体的には、インターロックフラグは、実行モジュール１８によってストア命令などの遅延命令における１サイクル目にセットされ、２サ

イクル目でリセットされる。各モジュールは、スケジューリングモジュール２３から１サイクルの実行指示１２３～１２６を受けたとき、インターロックフラグがセットされている場合はwait動作を行う。

【００６７】

＜フェッチモジュール＞

フェッチモジュール１４は、スケジューリングモジュール２３から実行指示１２６を受けたとき、ターゲットプロセッサのＩＦステージの１サイクルの動作をシミュレーションする。すなわち、フェッチモジュール１４は、メモリモジュール１２から、同時実行すべき複数命令（ここでは最大３命令とする）をフェッチし、フェッチ情報１１９としてフェッチ情報格納部１５に格納する。もし、フェッチ情報格納部１５の命令が全て有効であればフェッチ情報格納部１５にフェッチ情報を格納しない。これは、フェッチ情報格納部１５に格納されているデコードされていないフェッチ情報を更新しないためである。

【００６８】

フェッチ情報の一例を図１３に示す。同図においてフェッチ情報は、命令Ｘ～Ｚと、命令Ｘ～Ｚに対応する有効フラグ及び命令ＰＣとを含む。ここで、「命令Ｘ」はターゲットプロセッサでは命令デコーダ２０２に発行される命令、スロットｘに発行される命令ｘの命令コードである。同様に「命令Ｙ」、「命令Ｚ」は、それぞれ命令デコーダ２０３、２０４に発行される命令ｙ、ｚの命令コードである。それゆえ命令Ｘ、Ｙ、Ｚは命令アドレスの若い順になる。「有効フラグ」は、対応する命令が有効か否かを示す。同時実行すべき命令が３つの場合は命令Ｘ～Ｚの３つの有効フラグが、２つの場合は命令ＸとＹの２つ有効フラグが、単独で命令を実行すべき場合は命令Ｘの有効フラグが１（有効）になる。「命令ＰＣ」は、ターゲットプロセッサにおけるフェッチプログラムカウンタの内容に相当する命令アドレスを意味する。

【００６９】

フェッチ情報格納部１５は、図１３に示したフェッチ情報を格納するためのメモリ領域である。フェッチ情報は、フェッチモジュール１４とデコードモジュール１６とによって参照及び変更される。

【００７０】

＜デコードモジュール＞

デコードモジュール１６は、スケジューリングモジュール２３から実行指示１２５を受けたとき、インターロックフラグ１２７が１でなければ、ターゲットプロセッサのＤＣステージの１サイクルの動作をシミュレーションする。すなわち、デコードモジュール１６は、フェッチ情報格納部１５からフェッチ情報を読み出して解読し、解読結果をデコード情報１２０としてデコード情報格納部１７に格納する。その際、フェッチ情報格納部１５内のフェッチ情報に対して解読済命令の有効フラグを０（無効）にする。ただし、有効フラグが０の命令は、フェッチ情報中の当該命令の情報をそのままデコード情報に含める。もし、インターロックフラグが１であれば、デコード情報格納部１７のデコード情報を更新しない。

【００７１】

デコード情報の一例を図１４に示す。同図のようにデコード情報は、フェッチ情報と比べて、命令Ｘ～Ｙそれぞれに対応するレジスタ更新情報と、メモリアクセス命令と、有効フラグと、ＰＣと、レジスタ更新情報と、メモリアクセスアドレスと、メモリアクセスデータと、Ｒ／Ｗ情報とを新たに追加された点が主として異なっている。以下、フェッチ情報と同じ点は説明を省略して主に異なる点を説明する。

【００７２】

「命令ＰＣ」は、ターゲットプロセッサにおけるフェッチプログラムカウンタではなくデコードプログラムカウンタの内容に相当する命令アドレスを意味する。「命令Ｘのレジスタ更新情報」は、命令Ｘによって更新されるレジスタ（デスティネーションレジスタ）を示す。命令Ｙ、Ｚのレジスタ更新情報もそれぞれ同様である。この情報は出力依存の検出に用いられる。

【００７３】

同図中のメモリアクセス命令以下に列挙された各情報は、命令Ｘ～Ｚの何れかがメモリアクセス命令である場合にのみ有効であり、命令Ｘ～Ｚの何れもがメモリアクセス命令でない場合には無効である。「メモリアクセス命令」は、命令Ｘ

～Ｚの何れかと同じ命令であり、ターゲットプロセッサでは命令デコーダ２０２～２０４の何れかからマルチプレクサ２３０、２３１を介してメモリアクセス制御部２０９に発行される命令である。「有効フラグ」は、メモリアクセス命令が有効か否かを示し、デコードモジュール１６により初期値として１（有効）にセットされる。「命令ＰＣ」は、ターゲットプロセッサにおけるデコードプログラムカウンタの内容に相当する命令アドレスを意味する。「レジスタ更新情報」は当該メモリアクセス命令により更新されるレジスタを示す。「メモリアクセスアドレス」はアクセス先のメモリアドレスを示す。「Ｒ／W情報」は、ロード命令の場合はリード、ストア命令の場合はライト、メモリアクセス命令が存在しない場合はＮＯＰを示す。例えば、命令Ｚがロード命令（１ｄ　Ｒ０，（Ｒ１＋））の場合、命令Ｚの命令ＰＣとメモリアクセス命令の命令ＰＣは同じ値であるが、デコード情報中の命令Ｚには（Ｒ１＝Ｒ１＋４）を意味する操作コードが、メモリアクセス命令には（Ｒ０＝ｍｅｍ（Ｒ１））を意味する操作コードが設定される。命令Ｚのレジスタ更新情報はＲ１、メモリアクセス命令のレジスタ更新情報はＲ０となる。この場合、命令Ｚの操作コードとメモリアクセス命令の操作コードとは、演算機能を分担しており、ターゲットプロセッサのＡＬＵ２０７とメモリアクセス制御部２０９の演算機能の分担に対応している。

【００７４】

デコード情報格納部１７は、図１４に示したデコード情報を格納するためのメモリ領域である。デコード情報は、デコードモジュール１６によって書き込まれ、実行モジュール１８によって読み出される。

【００７５】

＜実行モジュール１８＞

実行モジュール１８は、スケジューリングモジュール２３から実行指示１２４を受けたとき、ターゲットプロセッサのＥＸ／ＭＥＭステージの１サイクルの動作をシミュレーションする。すなわち、実行モジュール１８は、デコード情報格納部１７からデコード情報１２０を読み出し、有効フラグが１（有効）の命令について、当該命令の演算内容のシミュレーション（具体的には当該命令に対応する命令実行関数を呼び出し）を行って第１レジスタファイルモジュール１１を更

新する。ここで、命令Ｘ、Ｙ、Ｚはこの順番でシミュレーションし、各命令のシミュレーションが終わる毎に実行モジュール１８は命令Ｘ、Ｙ、Ｚの何れの命令を実行したかを通知する命令実行通知１１２を出力する。

【００７６】

デコード情報中に遅延命令（例えば２サイクルのＭＥＭステージを要するメモリアクセス命令）が有効に存在する場合、実行モジュール１８は、遅延命令の遅延が残っていれば（つまり１サイクル目）、複数命令中の当該メモリアクセス命令以外の命令Ｘ～Ｚについてはシミュレーションを行い、メモリアクセス命令についてはシミュレーションを行わないで、共通情報格納部１３中のインターロックフラグを１にセットして当該サイクルのシミュレーションを終える。また、遅延命令の遅延が残っていなければ（つまり２サイクル目）、当該メモリアクセスのシミュレーションをメモリモジュール１２に対して行い、インターロックフラグをリセットする。その際、メモリアクセス命令がメモリライト命令である場合には、実行情報の一部とするためライト前のデータを読み出しておく。

【００７７】

このようなシミュレーションの結果、実行モジュール１８は実行情報を実行情報格納部１９に格納する。また、デコード情報格納部１７内デコード情報に対してシミュレーション済の命令の有効フラグを０（無効）にする。

【００７８】

実行情報の一例を図１５に示す。同図の実行情報は、図１４に示したデコード情報と比べて、ストア前メモリ内容が追加されている点が主として異なる。以下、デコード情報と同じ点は説明を省略して主に異なる点を説明する。

【００７９】

命令Ｘ～Ｙの「命令ＰＣ」及びメモリアクセス命令の「命令ＰＣ」は、ターゲットプロセッサにおける実行プログラムカウンタの内容に相当する命令アドレスを意味する。「ストア前データ」は、メモリアクセス命令がストア命令である場合にメモリライトする前のメモリデータであり、メモリ値退避部３２に出力され、メモリアクセス命令実行前の状態を復元するのに利用される。

【００８０】

実行情報格納部１９は、図１５に示した実行情報を格納するためのメモリ領域である。実行情報は、実行モジュール１８と完了処理モジュール２０とによって参照及び更新される。

【００８１】

＜完了処理モジュール＞

完了処理モジュール２０は、スケジューリングモジュール２３から実行指示１２３を受けたとき、インターロックフラグ１２７が１でなければ、ターゲットプロセッサのＷＢステージの１サイクルの動作をシミュレーションする。すなわち、完了処理モジュール２０は、実行情報格納部１９から実行情報を読み出し、有効フラグが１（有効）の命令について、ＷＢ（ライトバック）動作を行って、完了情報を完了情報格納部２１に格納する。また、実行情報格納部１９内の実行情報に対して完了した命令の有効フラグを０（無効）にする。

【００８２】

ただし、本実施例では、メモリアクセス命令以外の命令におけるレジスタへのライトバックは実行ステージにおいて完了しているので、ほとんどの命令が完了処理を必要としない。

【００８３】

完了情報の一例を図１６に示す。同図において完了情報は、図１５に示した実行情報と同様なので説明を省略する。ただし、「命令ＰＣ」はターゲットプロセッサにおける完了プログラムカウンタの内容に相当する命令アドレスを意味する。

【００８４】

完了情報格納部２１は、図１６に示した実行情報を格納するためのメモリ領域である。

過去状態更新制御部２２は、スケジューリングモジュール２３からの問い合わせ１１５に対して、共通情報格納部１３に格納されたインターロックフラグが１である場合には、第１レジスタファイルモジュール１１から第２レジスタファイルモジュール３１へのレジスタファイルのデータのコピーを禁止する旨の更新禁止通知１１６を応答し、インターロックフラグが０である場合には、コピーを禁

止しない旨の更新禁止通知１１６を応答する。ここで、第２レジスタファイルモジュール３１は、通常１サイクル前の第１レジスタファイルモジュール１１の内容が保存されている。過去状態更新制御部２２が禁止する旨を通知するのは、インターロックの発生によりＥＸ／ＭＥＭステージが２サイクルかかる場合に、第２レジスタファイルモジュール３１には、ＥＸ／ＭＥＭステージの開始前の状態のレジスタデータを保存しておくためである。

【００８５】

＜スケジューリングモジュール＞

スケジューリングモジュール２３は、シミュレーション実行指示１０１を受けたとき、複数命令を同時実行する１サイクル分のパイプライン処理をシミュレーションするようスケジューリングし、１サイクルのシミュレーション完了後に応答１０２を命令シミュレーション部３０に出力する。

【００８６】

図１７は、スケジューリングモジュール２３のスケジューリングによる命令群のシミュレーション処理を示すフローチャートである。

同図に示すように、スケジューリングモジュール２３は、シミュレーション制御部３４からシミュレーション実行指示１０１を受けたとき（Ｓ１１：yes）、過去状態更新制御部２２を呼び出すことにより問い合わせ１１５をかけ（Ｓ１２）、その応答として更新禁止通知１１６が第２レジスタファイルモジュール３１の更新を禁止していなければ（Ｓ１３：ｙｅｓ）第１レジスタファイルモジュール１１の内容を第２レジスタファイルモジュール３１にコピーし（Ｓ１４）、禁止していれば（Ｓ１３：ｎｏ）コピーしないで、実行指示１２３、１２４、１２５、１２６をこの順で出力する（Ｓ１５～１８）。実行指示１２３～１２６はシミュレーションプログラム中では関数呼び出しという形式で実現される。したがって、完了処理モジュール２０、実行モジュール１８、デコードモジュール１６、フェッチモジュール１４の順に実行されることになり、これにより命令群の１サイクル分のパイプライン処理がなされることになる。さらに、スケジューリングモジュール２３は、１サイクルのパイプライン処理が完了したことを通知する応答１０２をシミュレーション制御部３４に出力する。

【００８７】

＜命令シミュレーション部３０＞

第２レジスタファイルモジュール３１は、第１レジスタファイルモジュール１１の１サイクル以上前（つまり一命令群の実行前）のレジスタデータのコピーを保持する。この保持内容は、一命令群の実行後に当該命令群に含まれる各命令についての実行前のレジスタデータを復元するために用いられる。

【００８８】

メモリ値退避部３２は、実行モジュール１８においてストア命令のシミュレーションが実行される場合に、当該メモリ命令の書き込み先アドレスのストア前のメモリ値を退避させて保持する。

【００８９】

リソース情報変更部３３は、シミュレーション制御部３４から停止命令の通知を受けたとき、当該停止命令のシミュレーション実行前のリソースの状態を復元する。この停止命令は、パイプラインシミュレーション部１０によるシミュレーション実行済の最後の命令群に含まれる何れか１つの命令が指定される。リソース情報変更部３３は、パイプラインシミュレーション部１０による命令群のシミュレーション実行後のリソースと実行前のリソースとに基づいて、通知された停止命令がまだシミュレーションされていなかったとしたときのリソース（メモリデータ及びレジスタデータ）状態を復元する。言い換えれば、停止命令の直前の命令までシミュレーションがなされた場合に対応するリソース状態を復元する。ここで、命令群のシミュレーション実行後のリソースは第１レジスタファイルモジュール１１及びメモリモジュール１２に保持されている。また、当該命令群のシミュレーション実行前のリソースは第２レジスタファイルモジュール３１、メモリ値退避部３２に保持されている。

【００９０】

具体的には、停止命令が命令Ｘであるとき、リソース情報変更部３３は、命令群中の命令命令Ｘ、Ｙ及びＺのシミュレーション前の状態を復元する。停止命令が命令Ｙであるとき、リソース情報変更部３３は、命令群中の命令Ｙ及びＺのシミュレーション前の状態を復元し、停止命令が命令Ｚであるときは命令群中の命

令Ｚのシミュレーション前の状態を復元する。復元に際して、リソース情報変更部３３は、図１４に示した実行情報１２１を参照して、命令Ｘ、Ｙ、Ｚ、メモリアクセス命令のうち有効な命令によって更新されるリソースを判別し、命令Ｘ、Ｙ、Ｚ、メモリアクセス命令それぞれのシミュレーション前のリソースを取得する。取得したデータのうち停止命令の直前の状態を示すデータをレジスタデータ１３３やメモリ内容１３５としてデバッガ３ａに出力する。

【００９１】

シミュレーション制御部３４は、停止命令がどれであるか示す停止命令ポインタを保持し、デバッガ３ａからシミュレーション実行指示１３１に従って、命令単位のシミュレーション実行を制御し、デバッガ３ａにその結果として停止命令通知を送信する。すなわち、シミュレーション制御部３４は、パイプラインシミュレーション部１０による命令群のシミュレーションにおいて直前に実行済の複数命令の何れか１つを停止命令として管理しており、シミュレーション実行指示１３１を受けたとき、ブレーク条件に合致する命令がパイプラインシミュレーション部１０による直前にシミュレーション済の命令群に存在する場合には、停止命令ポインタをブレーク条件に合致する命令に更新し、停止命令通知１３２をデバッガ３ａ及びリソース情報変更部３３に出力する。一方、ブレーク条件に合致する命令がパイプラインシミュレーション部１０による直前にシミュレーション済の命令群に存在しない場合は、パイプラインシミュレーション部１０に命令群のシミュレーションを１サイクル進めるようにシミュレーション実行指示１０１を出力する。こうして、ブレーク条件に合致する命令が直前の命令群のシミュレーション結果に存在するようになるまで、シミュレーション実行指示１０１を出力する。

【００９２】

図１８は、シミュレーション制御部３４による命令単位のシミュレーション制御を示すフローチャートを示す。

同図では、シミュレーション制御部３４は、デバッガ３ａからシミュレーション実行指示１３１を受けると（Ｓ２１）、当該指示１３１がステップ実行指示であれば（ブレーク条件の指定がなければ）、ブレーク条件を次の「実行済みの命

令」と設定する（Ｓ２３）。次に、シミュレーション制御部３４は、現在の停止命令が命令Ｚに該当する場合には（Ｓ２４：ｙｅｓ）、スケジューリングモジュール２３にシミュレーション実行指示１０１を出力し、応答１０２の受信するまで待ち（Ｓ３０）、応答受信後にサイクル数を＋１インクリメントする。これにより、パイプラインシミュレーション部１０において新たに次の命令群のシミュレーションがなされる。また、このサイクル数は、命令単位のシミュレーションのサイクル数でなく、命令群単位のシミュレーションにおけるサイクル数である。これにより、シミュレーション制御部３４は、ターゲットプロセッサにおけるサイクル数を正確にカウントしている。

【００９３】

さらに、シミュレーション制御部３４は、シミュレーションされた新たな命令群における命令Ｘが実行済で且つブレーク条件に合致するか否かを判定する（Ｓ３２）。命令Ｘが実行済であるか否かの判定については、シミュレーション制御部３４が実行情報格納部１９に格納された実行情報中の命令Ｘの有効フラグが１（有効）でありかつ共通情報格納部１３に格納されたインターロックフラグが０（インターロック中でない）であれば実行済と判定する。有効フラグが０（無効）であれば、当該命令群中に有効な命令Ｘが存在しないし、インターロックフラグが１（インターロック中）であれば、命令Ｘが存在していても命令群のシミュレーションが完了していないからである。命令Ｙ、Ｚが実行済か否かの判定についても同様である。

【００９４】

Ｓ３２の判定の結果、命令Ｘが実行済で且つブレーク条件に合致する場合には、停止命令ポインタを命令Ｘに更新し（Ｓ３３）、シミュレーション制御部３４は、リソース情報変更部３３及びデバッガ３ａに更新された停止命令を停止命令通知１３２として通知する（Ｓ３４）。Ｓ３２の判定の結果、命令Ｘが実行済でないか、またはブレーク条件に合致しない場合には、Ｓ２６に進む。

【００９５】

また、シミュレーション制御部３４は、現在の停止命令が命令Ｙであり（Ｓ２５：ｙｅｓ）、命令Ｚが実行済で且つブレーク条件に合致するか否かを判定する

（Ｓ２８）。Ｓ２８の判定の結果、命令Ｚが実行済で且つブレーク条件に合致する場合には、停止命令ポインタを命令Ｚに更新し（Ｓ２９）、リソース情報変更部３３及びデバッガ３ａに更新された停止命令を停止命令通知１３２として通知する（Ｓ３４）。Ｓ２８の判定の結果、命令Ｚが実行済でない、またはブレーク条件に合致しない場合には、Ｓ３０に進む。

【００９６】

また、シミュレーション制御部３４は、現在の停止命令が命令Ｘであるとき（Ｓ２５で命令Ｙでないと判定されたとき）、命令Ｙが実行済で且つブレーク条件に合致するか否かを判定する（Ｓ２６）。Ｓ２６の判定の結果、命令Ｙが実行済で且つブレーク条件に合致する場合には、停止命令ポインタを命令Ｙに更新し（Ｓ２７）、リソース情報変更部３３及びデバッガ３ａに更新された停止命令を停止命令通知１３２として通知する（Ｓ３４）。Ｓ２６の判定の結果、命令Ｙが実行済でない、またはブレーク条件に合致しない場合には、Ｓ３８に進む。

【００９７】

このように、シミュレーション制御部３４は、ブレーク条件に合致する命令が命令群のシミュレーション結果中に見つかるまで、順次パイプラインシミュレーション部１０に命令群の１サイクルのシミュレーション実行指示１０１を出力する。したがって、停止命令が見つかった時点では、その停止命令の属する命令群のシミュレーション後とそのシミュレーション前の状態（リソース）が必ず保存されていることになる。これにより命令単位にシミュレーション実行前後の状態の復元を可能にしている

【００９８】

図１９は、リソース情報変更部３３の構成を示すブロック図である。同図のようにリソース情報変更部３３は通常命令結果生成部３５、第１補完部３６、第２補完部３７、メモリ内容選択部３８を備え、シミュレーション制御部３４から通知される停止命令のシミュレーション実行前の状態を復元する。

【００９９】

通常命令結果生成部３５は、シミュレーション制御部３４から停止命令通知１３２を受けたとき、第１レジスタファイルモジュール１１、メモリモジュール１

２、第２レジスタファイルモジュール３１、メモリ値退避部３２の内容に基づいて当該停止命令の実行前のレジスタファイルを復元する。通常命令結果生成部３５は、通知された命令が通常命令であっても遅延命令であってもレジスタファイルを復元するが、遅延命令である場合や出力依存がある場合には、第１補完部３６、第２補完部３７によって復元を補完される。

【０１００】

通常命令結果生成部３５の詳細な構成を示すブロック図を図２０に示す。同図のように通常命令結果生成部３５は、命令毎レジスタファイル退避部３９と、命令Ｘのシミュレーション実行結果を保持するための第３レジスタファイルモジュール４０と、命令Ｙのシミュレーション実行結果を保持するための第４レジスタファイルモジュール４１を備える。

【０１０１】

命令毎レジスタファイル退避部３９は、実行モジュール１８から出力されためいれいＸの命令実行通知を受け、第１レジスタファイルモジュール１１の内容を、第３レジスタファイルモジュール４０に命令Ｘまで実行した場合のシミュレーション実行結果としてコピーし、同様に命令Ｙの命令実行通知を受け、第１レジスタファイル１１の内容を第４レジスタファイルモジュール４１に命令Ｙまで実行した場合のシミュレーション実行結果としてコピーする。これにより命令Ｘ、Ｙ、Ｚを順次実行した場合の各レジスタデータが第３、第４、第１レジスタファイルモジュールに格納されることになる。このとき、第２レジスタファイルモジュール３１は、直前の命令群の実行結果を格納している。また、命令毎レジスタファイル退避部３９は、シミュレーション制御部３４から停止命令の通知を受けたとき、停止命令が命令Ｘであればその直前の命令群の実行結果を示す第２レジスタファイルモジュール３１の内容を出力する。同様に、停止命令が命令Ｙであれば第３レジスタファイルモジュール４０の内容を、停止命令が命令Ｚであれば第４レジスタファイルモジュール４１の内容を第１補完部３６、第２補完部３７を介してデバッガ３ａに出力する。この出力は、命令Ｘ～Ｚが通常命令（遅延を生じさせない命令）である場合は、第１補完部３６、第２補完部３７による補完を受ける必要がなく、そのままデバッガ３ａに出力される。

【０１０２】

このように、通常命令結果生成部３５は、命令群に遅延命令が含まれない場合には、停止命令が命令Ｘ、Ｙ、Ｚのどれであっても当該停止命令直前の命令が実行済の時点におけるレジスタファイルの内容を生成する。それゆえ、ターゲットプロセッサが同一命令郡中の出力依存を許さないアーキテクチャの場合、第３、第４レジスタファイルモジュール４０、４１は省略可能である。

【０１０３】

第１補完部３６は、ＥＸステージで遅延命令（メモリアクセス命令）が命令群に含まれている場合に、実行情報格納部１９から、当該遅延命令が更新するレジスタ番号と更新内容（メモリアクセスデータ）を参照し、通常命令結果生成部３５によって生成された停止命令実行前の状態を示すレジスタファイルの内容を補完する。

【０１０４】

例えば、ロード命令（ｌｄ　Ｒ０，（Ｒ１＋））が２サイクルかかるものとすると、１サイクル目ではレジスタＲ１の更新が完了し、２サイクル目ではレジスタＲ０の更新が完了しなければならない。なぜならターゲットプロセッサにおいてＲ１の更新は１サイクルで実行する３つのＡＬＵの何れかが分担し、Ｒ０の更新は２サイクルを要するメモリアクセス制御部２０９が分担するからである。そのため、実行情報格納部１９は、メモリアクセス命令の更新レジスタ番号と、更新内容であるメモリアクセスデータを保持する。第１補完部３６は、停止命令が同一命令群のロード命令よりも後の命令であった場合、実行情報格納部１９からメモリアクセス命令のレジスタ更新情報と更新される内容であるメモリアクセスデータとを読み出し、更新されるレジスタを遅延レジスタとして、メモリアクセスデータを遅延データとして認識する。

【０１０５】

さらに、第１補完部３６は、通常命令結果生成部３５から出力されたレジスタファイルの内容の遅延レジスタに該当する箇所を遅延データで更新することにより遅延命令が存在する場合のレジスタファイルを復元する。ただし、ロード命令で更新されるレジスタが、同一命令郡中のより後ろの命令により更新される場合

（出力依存がある場合）はロード命令によるレジスタ書き込みがキャンセルされるので補完を行わない。また、遅延命令がレジスタを更新しない命令（ストア命令など）の場合も当然補完を行わない。これにより命令単位のシミュレーションを指示された場合に命令単位でリソースを正確に復元する。

【０１０６】

第２補完部３７は、ＭＥＭステージで２サイクル以上を要する遅延命令と、遅延より後に配置された遅延命令と出力依存関係にあるその他の命令（以降出力依存命令と呼ぶ）が同一命令群に含まれる場合において、停止命令より前に遅延命令があり、停止命令以降に出力依存命令がある場合に、第１補完部３６と同様に遅延レジスタを遅延データで更新し、第１補完部３６から出力されるレジスタファイルの内容を補完する。

【０１０７】

例えば、命令Ｙがロード命令（ｌｄ　Ｒ１，（Ｒ２＋））、命令Ｚが転送命令（ｍｏｖ　Ｒ１，３）であって、ロード命令のＥＸステージが２サイクルを要するものとする。この場合、命令Ｙと命令Ｚはともにレジスタ Ｒ１をデスティネーションとする出力依存関係を有するが、命令Ｚによるレジスタ Ｒ１の更新がなされなければならない。なぜならターゲットプロセッサでは、ＥＸステージの１サイクル目においてロード命令によるレジスタＲ２のインクリメントとｍｏｖ命令によるレジスタＲ１の更新がなされ、２サイクル目においてロード命令によるレジスタＲ１の更新が命令キャンセル部２２０によってキャンセルされるからである。ところが、命令単位のシミュレーションにおいては、まだ実行もされていない命令Ｚによって命令Ｙの実行結果がキャンセルされるのは、不自然である。むしろ、命令Ｚによりレジスタ Ｒ１が上書きされたことによりキャンセルされたものとするべきであろう。そのため、第２補完部３７は、命令Ｙ、Ｚを順番に１つずつ実行した場合と同じ結果を生成するため、出力依存によりキャンセルされる実行内容であっても、その実行結果を補完する。

【０１０８】

図２１は、メモリ値退避部３２の詳細な構成を示すブロック図である。同図のようにメモリ値退避部３２は、ストア命令によって書き込まれる前のメモリデー

タを保持するためのストア前データ格納部４２と、当該ストア命令によって指定されたアドレスを保持するストアアドレス格納部４３と、メモリ内容変更部４４とを備える。

【０１０９】

メモリ内容変更部４４は、リソース情報変更部３３から停止命令通知を受けたとき、命令群中の停止命令以降にストア命令が含まれている場合に、当該ストア命令実行前のメモリ内容を復元する。シミュレーション制御部３４から読み出しを指定されたメモリアドレス及びサイズ１０４に、当該ストア命令のストア先のアドレスが含まれている場合には、メモリモジュール１２のデータの代わりにストア前データ格納部４２のデータをメモリ内容１０５に含ませてリソース情報変更部３３に出力する。

以上のように構成された実施の形態におけるシミュレーションシステム１について、プログラム例を挙げて説明する。

【０１１０】

＜第１プログラム例＞

図２２は、シミュレーション対象の第１のプログラム例を示す図である。同図のプログラム例は、同時実行される命令６、７からなる命令群１と命令８からなる命令群２のみを示している。各命令について命令アドレスを示す「ＰＣ」と「ニーモニック」と「シミュレーション結果」と「表示結果」と「停止」とを記している。命令群１のシミュレーション直前の状態は、｛Ｒ０、Ｒ１、Ｒ２、Ｒ３、Ｆ０｝＝｛１、０，０，０、１｝であるものとする。

【０１１１】

「シミュレーション結果」は、命令シミュレーション部３０による命令単位で当該命令をシミュレーションした後の状態（同図ではＲ０～Ｒ３、Ｆ０のみを記した）を示す。「表示結果」は、当該命令が停止命令であるときにデバッガ３に表示される状態であって、当該停止命令の実行前の状態を示す。「停止」は、当該命令をブレーク条件とした場合にブレーク（停止）するか否か、つまり当該命令が停止命令となるか否かを示す。

【０１１２】

このプログラム例の場合、命令６、８は停止命令となり得るが、命令７は停止命令にはならない。なぜなら、命令７はフラグＦ０により無効化されるので、命令７をブレーク条件とするシミュレーションを行っても、ブレークしないことになる。その際、シミュレーションシステム１は、停止命令の検出を、命令のシミュレーション実行前にブレーク条件を判定するのではなく、命令群のシミュレーション実行後にブレーク条件を判定しているからである。これによりターゲットプロセッサにおけるプログラムの実行経路（又は分岐経路）と同じ実行経路（又は分岐経路）をシミュレーションした結果を得ている。

【０１１３】

より詳しく言うと、パイプラインシミュレーション部１０での命令群１のシミュレーションにおいて、実行モジュール１８は、命令６のシミュレーションによりフラグＦ０がリセットされるので、命令７のシミュレーションをキャンセルし、命令７の有効フラグをリセットした実行情報１２２を実行情報格納部１９に格納する。シミュレーション制御部３４は、命令群１のシミュレーション結果のうち有効フラグが０ならば命令がまだ実行されていない（または命令が存在しないと）と判断するので、命令７についてブレーク条件の合致するかどうか判定を行わない。それゆえ、命令７は停止命令とはならない。

【０１１４】

このようにシミュレーションシステム１は、命令群単位にシミュレーションを停止するのではなく、命令単位に停止することができる。しかも、ターゲットプロセッサがキャンセル機構を備える場合には、命令群中の命令キャンセルも正確にシミュレーションする。

【０１１５】

＜第２プログラム例＞

図２３は、シミュレーション対象の第２のプログラム例を示す図である。同図のプログラム例では、同時実行される命令１～３からなる命令群１と命令４、５からなる命令群２を示している。各命令についての「ＰＣ」「ニーモニック」「表示結果」「停止」は図２２と同様である。ただし、命令群１のシミュレーション直前の状態は、｛Ｒ０、Ｒ１、Ｒ２、Ｒ３、Ｒ４、Ｒ５、Ｒ６｝＝｛１０、５

，０，０、０、１、２｝、ｍｅｍ（０）＝１００、ｍｅｍ（４）＝２００であるものとする。

【０１１６】

このプログラム例中の命令３は、レジスタＲ４の内容をアドレスとしてメモリデータを読み出してレジスタＲ３にロードし、さらにレジスタＲ４を＋４インクリメントすることを指示する命令である。命令４は、レジスタＲ２のデータを、レジスタＲ４の内容をアドレスとしてメモリにストアし、さらにレジスタＲ４を＋４インクリメントすることを指示する命令である。この命令３と命令４とはデータ依存関係がある。つまり、命令３がレジスタＲ４を＋４インクリメントした結果を命令４は使用しているので、命令３の実行完了後でなければ命令４は正しく実行できない。

【０１１７】

この点、ターゲットプロセッサでは、図１０で説明したフォワーディングにより、インターロックの発生を防止している。すなわち、命令３の実行ステージにおいて＋４インクリメントしたレジスタＲ４のデータを、フォワーディングユニット２５３により次のサイクルで命令４の実行ステージに供給している。図１０ではロード命令の実行ステージが２サイクルかかる場合を前提にしているので、１サイクルのインターロックが発生しているけれども、このインターロックはデータ依存関係に起因するのではなく、２サイクルの実行ステージに起因する。ロード命令の実行ステージが１サイクルだけ場合には、図１０においてもインターロックは発生しない。

【０１１８】

これに対応するシミュレーションシステム１のシミュレーションについて、（１）命令３のロード命令が１サイクルで終了する場合、（２）命令３のロード命令が２サイクルを要する場合に分けて説明する。

【０１１９】

（１）命令３（ロード命令）の実行ステージが１サイクルで終了する場合

実行モジュール１８は、命令３の実行ステージのシミュレーションにおいて第１レジスタファイルモジュール１１のレジスタＲ４のデータを格納する。さらに

、実行モジュール１８は、次のサイクルで命令４の実行ステージのシミュレーションとして第１レジスタファイルモジュール１１のレジスタＲ４を用いて命令４のシミュレーションを行う。この場合、シミュレーションシステム１は、ターゲットプロセッサのフォワーディングとに相当する機能を第１レジスタファイルモジュール１１を用いて実現している。

【０１２０】

一方、ブレーク条件又はステップ実行により命令４を停止命令としている場合に、シミュレーション制御部３４は、図１８に示したフローの通り、パイプラインシミュレーション部１０における命令群２のシミュレーション結果において命令Ｘが命令４に合致するので、停止命令を命令Ｘに更新してデバッガ３ａ及びリソース情報変更部３３にその旨通知する。リソース情報変更部３３は、命令Ｘの実行前の状態を復元してデバッガ３ａにレジスタデータ１３３及びメモリ内容１３５を通知する。このようにして命令Ｘ（命令４）の直前の命令３のシミュレーション結果は、図２３における命令４の表示結果欄のようになる。

【０１２１】

また、この状態でｍｅｍ（４）をデバッガ３ａが参照した場合、命令４（ストア命令）の実行前のｍｅｍ（４）を表示するためリソース情報変更部３３はメモリ値退避部３２に退避されている値２００を表示する。

【０１２２】

（２）命令３（ロード命令）の実行ステージが２サイクルかかる場合

この場合も、（１）と同様に、図２４に示した表示結果を得るが、シミュレーションシステム１において２サイクルのシミュレーションを行う点が（１）と異なる。これは、ターゲットプロセッサのサイクル数についても正確にシミュレーションするためである。

【０１２３】

詳しくは、命令３はデコード情報格納部１７中のデコード情報１２０においてレジスタＲ４に対する＋４のインクリメントを指示する「命令Ｚ」及びＲ４をアドレスとしてメモリデータを読み出しＲ３に格納する「メモリアクセス命令」として設定される。実行モジュール１８は、命令３の実行ステージの１サイクル目

のシミュレーションにおいて、当該「命令Ｚ」（レジスタＲ４の＋４インクリメント）のシミュレーションとして第１レジスタファイルモジュール１１のレジスタＲ４を更新するすると共にインターロックフラグをセットする。２サイクル目で当該「メモリアクセス命令」をシミュレーションする。

【０１２４】

さらに、命令３の１サイクル目の次のサイクルでは、命令Ｚのシミュレーション結果（第１レジスタファイルモジュール１１の更新されたレジスタＲ４）が命令４により利用可能になる。これによりフォワーディングと同様の機能を果たしている。

【０１２５】

また、一方、ブレーク条件又はステップ実行により命令４を停止命令としている場合に、シミュレーション制御部３４は、図１８に示したフローに従って、（１）と同様に、命令Ｘの実行前の状態を復元してデバッガ３ａにレジスタデータ１３３及びメモリ内容１３５を通知する。

【０１２６】

上記の（１）（２）はシミュレーション制御部３４によって命令単位に停止命令を決定し、リソース情報変更部３３によって決定された停止命令が未実行で直前の命令のシミュレーション済の状態を復元するので、同じ結果となる。また、パイプラインシミュレーション部１０ではサイクル単位に命令群のシミュレーションを行っているので、シミュレーション制御部３４にカウントされるサイクル数は、ターゲットプロセッサにおけるサイクル数を（１）の場合も（２）の場合もそれぞれ正確に正確にカウントすることができる。

【０１２７】

また、図２３における命令４（ストア命令）の実行ステージが１サイクルである場合は（１）と、２サイクルである場合は（２）と同様にして、シミュレーション制御部３４はサイクル数を正確にカウントすることができる。

【０１２８】

＜第３プログラム例＞

図２４は、遅延命令が出力依存関係を有している場合のシミュレーション対象

の第３のプログラム例を示す図である。同図のプログラム例では、命令群５に属する命令１２～１４に加えて、パイプラインシミュレーション部１０による命令群５のシミュレーション後の第１レジスタファイルモジュール１１、第２レジスタファイルモジュール３１、第３レジスタファイルモジュール４０、第４レジスタファイルモジュール４１の各内容（同図ではＲ０～Ｒ２のみ）と、実行情報格納部１９内のメモリアクセスデータの内容を記してある。

【０１２９】

このプログラム例では、命令１３が遅延命令であり、かつ命令１３と命令１４とが出力依存関係を有している。命令群５のシミュレーション直前の状態は｛Ｒ０、Ｒ１、Ｒ２｝＝｛０，０，０｝ｍｅｍ（０）＝２００であるものとする。また、命令１３のメモリアクセスは２サイクルを要するものとする。

【０１３０】

パイプラインシミュレーション部１０及びシミュレーション制御部３４の動作は、第２プログラム例の（２）と同様に、遅延命令のよる２サイクルの実行ステージをシミュレーションする。リソース情報変更部３３は、出力依存関係のある命令１３、１４については、あたかも順に実行したような結果を生成する。つまり、リソース情報変更部３３は、同図の命令１４の「表示結果」欄のように、直前の命令１３のシミュレーション結果を生成する。

【０１３１】

この点、ターゲットプロセッサでは、命令１３によるレジスタＲ１の更新はキャンセルされ、命令１４によるレジスタＲ１の更新のみ実行される。これは命令１２、１３の順に実行した結果を得るためである。

【０１３２】

これに対して、リソース情報変更部３３は、命令１３の次の命令（命令１４）が停止命令であれば、命令１３のシミュレーション結果を生成し、命令１４の次の命令が停止命令であれば、命令１４のシミュレーション結果を生成する。リソース情報変更部３３が命令１２、１３の順に実行した結果を得るという点では、ターゲットプロセッサと同じであるが、リソース情報変更部３３はキャンセルされることになる命令１３についてもシミュレーション結果を生成する点で異なる

。デバッガ３ａのユーザにとってはデータ依存関係のある命令が結果的にキャンセルされる過程がわかる点で、使い勝手がよくなっている。

【０１３３】

具体的には、パイプラインシミュレーション部１０は、命令群５のシミュレーションによって、第１、第２レジスタファイルモジュール１１、３１を更新する。またこのとき、レジスタファイル退避部３９は実行モジュール１８から命令実行通知を受けて、第３、４レジスタファイルモジュール４０、４１を更新する。

【０１３４】

その結果、第１レジスタファイルモジュール１１は命令群５のシミュレーション直後のデータを保持する。これは、命令Ｚ（つまり命令１４）のシミュレーション直後のデータである。第２レジスタファイルモジュール３１は命令群５のシミュレーション直前のデータを保持する。第３レジスタファイルモジュール４０は命令Ｘ（命令１２）のシミュレーション後のデータを、第４レジスタファイルモジュール４１は命令Ｙ（つまり命令１３）シミュレーション後のデータを保持する。実行情報格納部１９内のメモリアクセスデータは、命令１３でロードされたメモリの内容を保持する。

【０１３５】

命令１４が停止命令として指示されたとき、リソース情報変更部３３は、第４レジスタファイルモジュール４１のデータに対して第１補完部３６でメモリアクセスデータを補完したデータをレジスタデータ１３３としてデバッガ３ａに出力する。

【０１３６】

このように、シミュレーションシステム１によれば、遅延命令と他の命令とに出力依存関係がある場合、それらの命令を順に実行した命令単位のシミュレーション結果を得ることができ、しかも、命令群単位のサイクル数を正確にカウントすることができる。

【０１３７】

以上説明してきたように本発明の実施の形態におけるシミュレーションシステム１によれば、複数命令を同時実行するプロセッサをターゲットとするが、命令

単位にシミュレーションを行うことができる。それゆえ、命令群単位のブレークではなく、同時実行される命令単位にブレークすることができる。

【０１３８】

しかも、シミュレーションシステム１は、命令群のサイクル単位のシミュレーションと命令単位のシミュレーションとからなる２段階のシミュレーションを行うので、ターゲットプロセッサのサイクル数を正確にシミュレーションすることができる。

【０１３９】

また、ターゲットプロセッサがフォワーディング機構を有している場合、遅延命令によりインターロックが発生する場合、およびキャンセル機構を有している場合でもターゲットプロセッサのサイクル数を正確にシミュレーションすることができる。

【０１４０】

なお、上記実施の形態において、シミュレーション制御部３４は、命令単位に命令のシミュレーション後にブレーク条件に合致するか否かを判定しているが、この代わりに、シミュレーション前にブレーク条件を判定するように構成してもよい。この場合に、図２２の第１プログラム例に対するシミュレーション結果及び表示結果を図２５に示す。図２２と比べて「表示結果」欄のみが異なっている。つまり図２５の「表示結果」欄は「シミュレーション結果」欄と同じく当該命令のシミュレーション結果を示している。図２５では、キャンセルされる命令７をブレーク条件とした場合に、停止することになる。この場合、ソフトウェア開発者は、命令７がキャンセルされるか否かを確認することができる。

【０１４１】

また、シミュレーション制御部３４は、命令単位のシミュレーション結果をデバッガ３ａに出力しているが、これに加えて、パイプラインシミュレーション部１０では命令群をサイクル単位でシミュレーションしているので、命令群単位のシミュレーション結果又は命令群のサイクル単位のシミュレーション結果をデバッガ３ａに出力するようにしてもよい。また、命令単位のシミュレーション結果と、命令群単位のシミュレーション又は命令群のサイクル単位のシミュレーショ

ンとを選択的に切り替える構成としてもよい。

【０１４２】

また、上記実施の形態においてメモリアクセス命令のＭＥＭステージの所要サイクル数は、１サイクル以上の如何なるサイクル数であっても、また、動的に変化する場合であっても、本発明のシミュレーション装置を適用してターゲットプロセッサにおけるサイクル数を正確にシミュレーションすることができる。この場合は、メモリモジュールにおいてメモリアクセスに対する応答（ACK）がどのサイクルで返るかをシミュレーションする構成とすればよい。

【０１４３】

第３レジスタファイルモジュール４０は、全レジスタのデータを保持せず、命令により更新されるレジスタのみのデータを保持するようにしてもよい。第４レジスタファイルモジュール４１についても同様である。

【０１４４】

【発明の効果】

以上説明してきたように本発明のシミュレーション装置によれば、複数命令を同時実行するプロセッサをターゲットとし、命令単位にシミュレーションを行うことができる。それゆえ、命令群単位のブレークではなく、同時実行される命令単位にブレークすることができる。また、ソフトウェア開発者に対しては、あたかも命令を１つずつ逐次シミュレーションしているかのように見せている。

【０１４５】

しかも、シミュレーション装置は、命令群のサイクル単位のシミュレーションと命令単位のシミュレーションとからなる２段階のシミュレーションを行うので、ターゲットプロセッサのサイクル数を正確にシミュレーションすることができる。

【０１４６】

また、ターゲットプロセッサがフォワーディング機構を有している場合や、遅延命令によりインターロックが発生する場合、キャンセル機構を有している場合でもターゲットプロセッサのサイクル数を正確にシミュレーションすることができる。

【図面の簡単な説明】

【図１】

本発明の実施の形態におけるシミュレーションシステム１の外観を示す図である。

【図２】

シミュレーション装置２においてシミュレーションソフトウェア起動後の表示内容の一例を示す。

【図３】

デバッグ装置３においてデバッグソフトウェア実行中の表示内容の一例を示す。

【図４】

シミュレーション装置の対象となるターゲットプロセッサの構成を示す図である。

【図５】

ターゲットプロセッサのパイプラインステージの動作タイミングを示す図である。

【図６】

パイプライン処理される命令群の一例を示す図である。

【図７】

パイプラインのサイクル数を説明する図である。

【図８】

フォワーディングを伴うパイプラインステージの動作タイミングを示す図である。

【図９】

命令がキャンセルされる場合の他の命令列の例を示す図である。

【図１０】

フォワーディングを伴うパイプライン処理を示す説明図である。

【図１１】

シミュレーション装置の対象となる他のターゲットプロセッサの構成を示す図

である。

【図１２】

シミュレーションシステムの構成を示す機能ブロック図である。

【図１３】

フェッチ情報の一例を示す図である。

【図１４】

デコード情報の一例を示す図である。

【図１５】

実行情報の一例を示す図である。

【図１６】

完了情報の一例を示す図である。

【図１７】

スケジューリングモジュールよる命令群のシミュレーション処理を示すフローチャートである。

【図１８】

シミュレーション制御部による命令単位のシミュレーション制御を示すフローチャートを示す。

【図１９】

リソース情報変更部の構成を示すブロック図である。

【図２０】

通常命令結果生成部の詳細な構成を示すブロック図である。

【図２１】

メモリ値退避部の詳細な構成を示すブロック図である。

【図２２】

シミュレーション対象の第１のプログラム例を示す図である。

【図２３】

シミュレーション対象の第２のプログラム例を示す図である。

【図２４】

シミュレーション対象の第３のプログラム例を示す図である。

【図２５】

シミュレーション対象の第４のプログラム例を示す図である。

【符号の説明】

１　シミュレーションシステム

２　シミュレーション装置

２ａ　本体装置

２ｂ　表示装置

２ｃ　入力装置

３　デバッグ装置

３ａ　デバッガ

３ｂ　表示装置

３ｃ　入力装置

４　ユーザインターフェース

１０　パイプラインシミュレーション部

１１　レジスタファイルモジュール

１２　メモリモジュール

１３　共通情報格納部

１４　フェッチモジュール

１５　フェッチ情報格納部

１６　デコードモジュール

１７　デコード情報格納部

１８　実行モジュール

１９　実行情報格納部

２０　完了処理モジュール

２１　完了情報格納部

２２　過去状態更新制御部

２３　スケジューリングモジュール

３０　命令シミュレーション部

３１　レジスタファイルモジュール

３２　メモリ値退避部

３３　リソース情報変更部

３４　シミュレーション制御部

３４　シミュレーション制御部

３５　通常命令結果生成部

３６　補完部

３７　補完部

３８　メモリ内容選択部

３９　命令毎レジスタファイル退避部

４０　レジスタファイルモジュール

４１　レジスタファイルモジュール

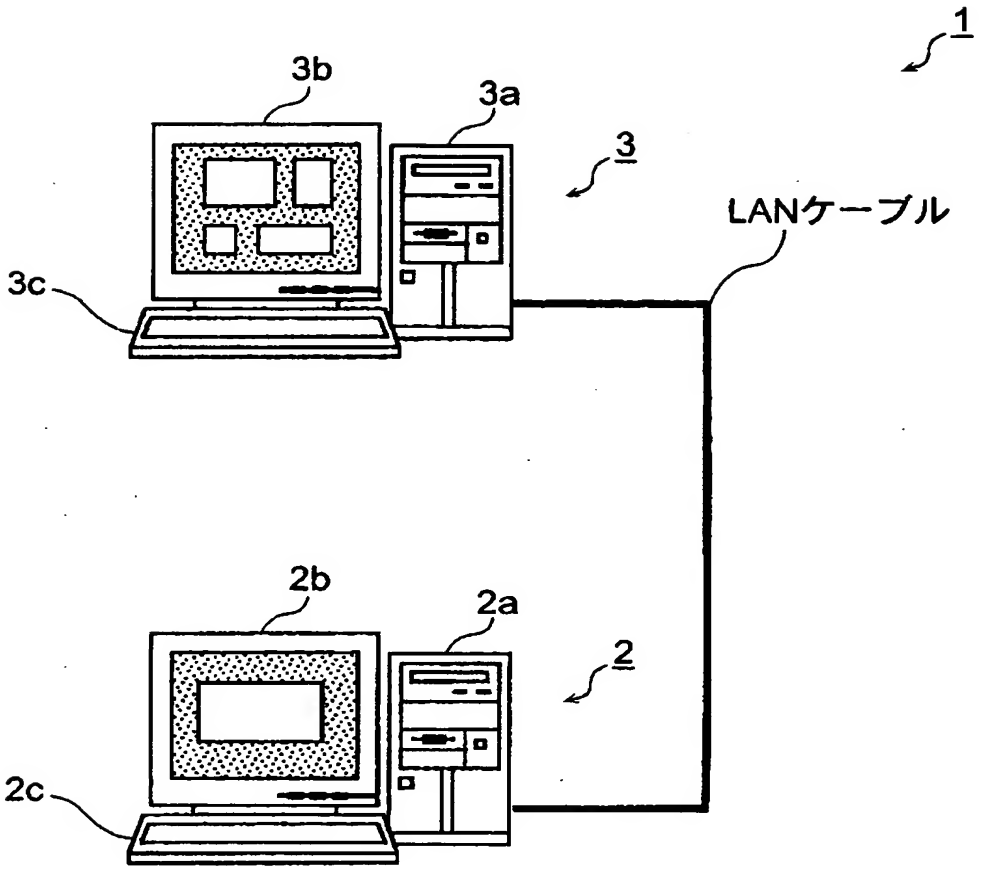
４２　ストア前データ格納部

４３　ストアアドレス格納部

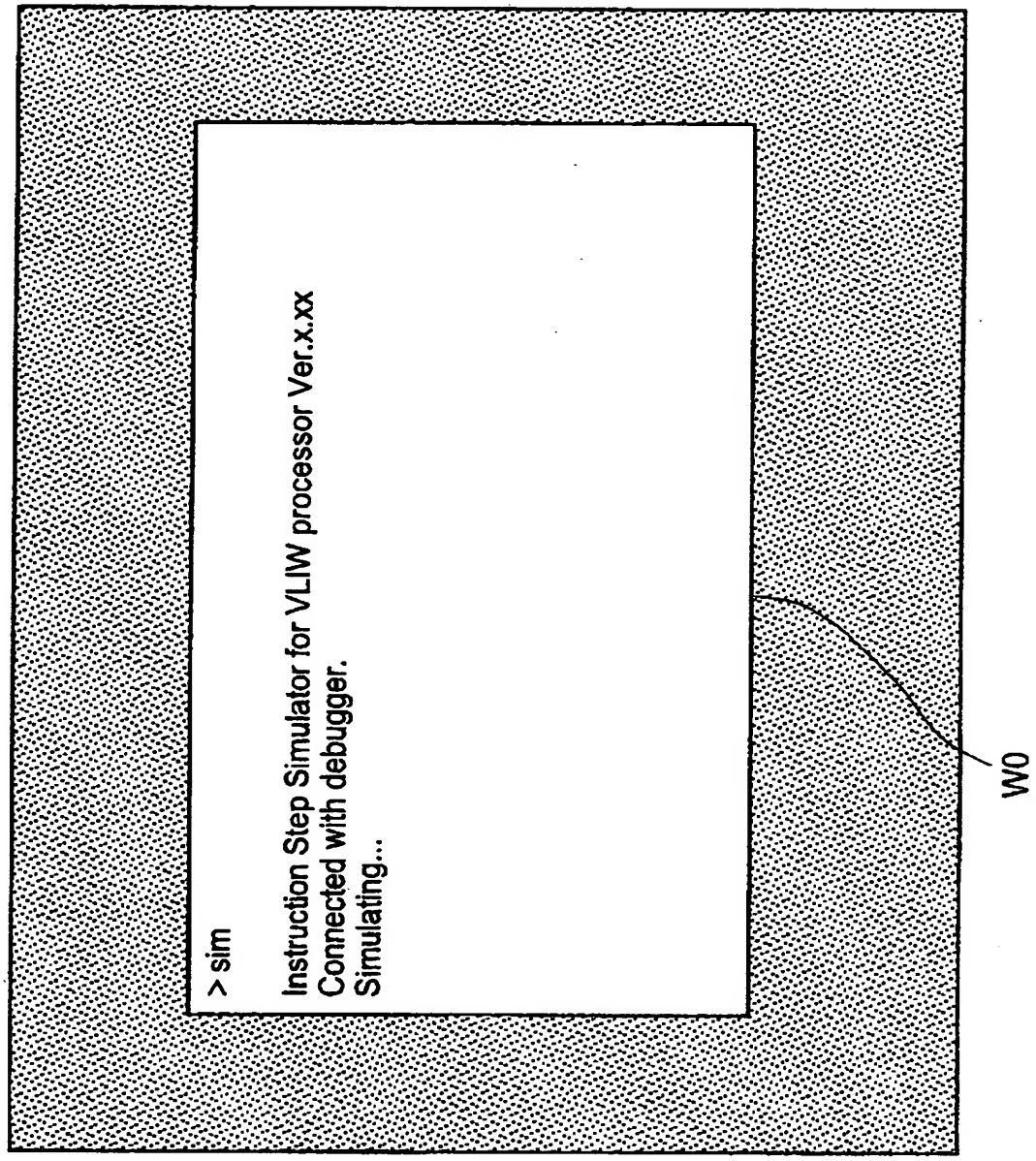
４４　メモリ内容変更部

【書類名】　　図面

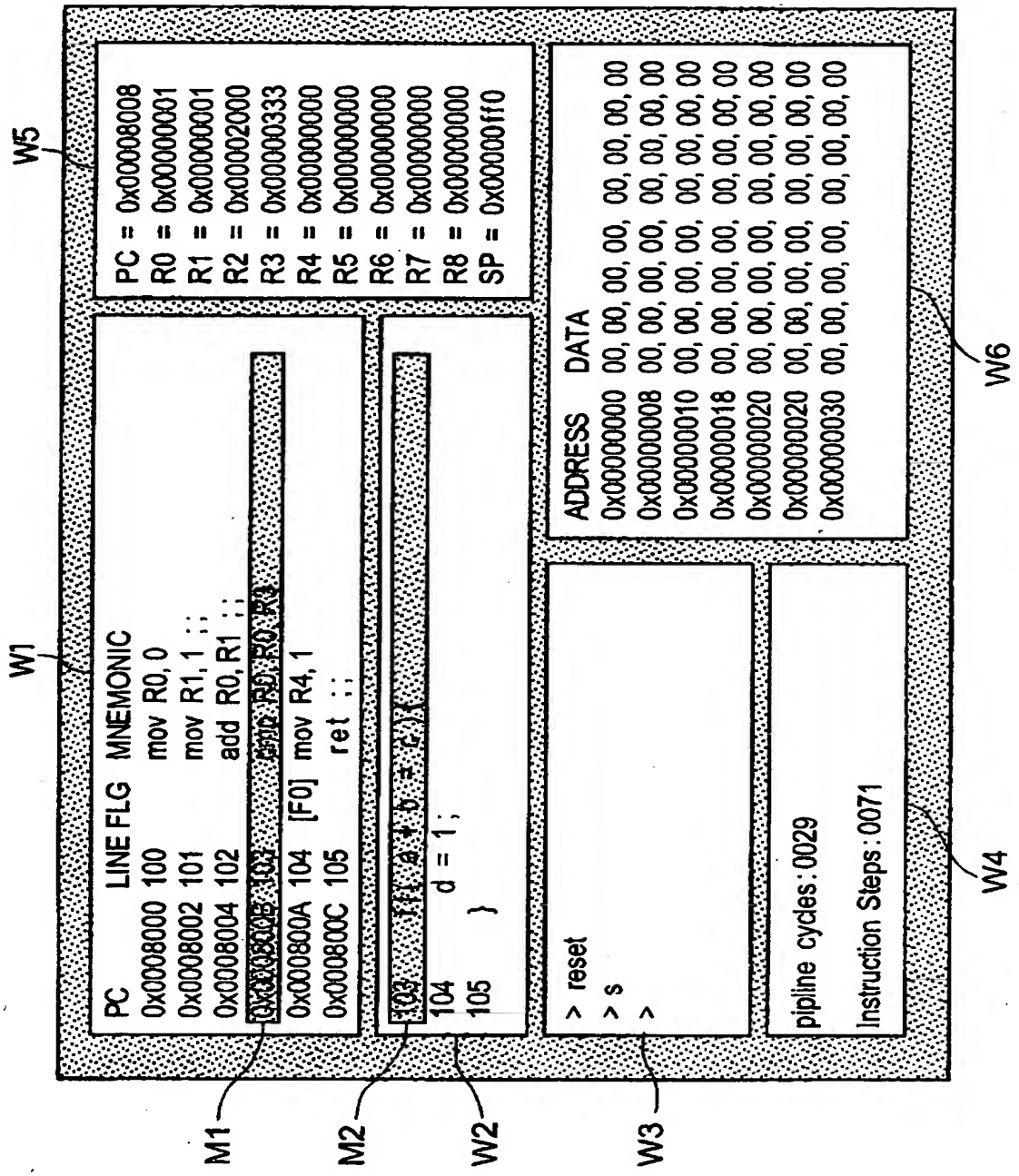
【図１】



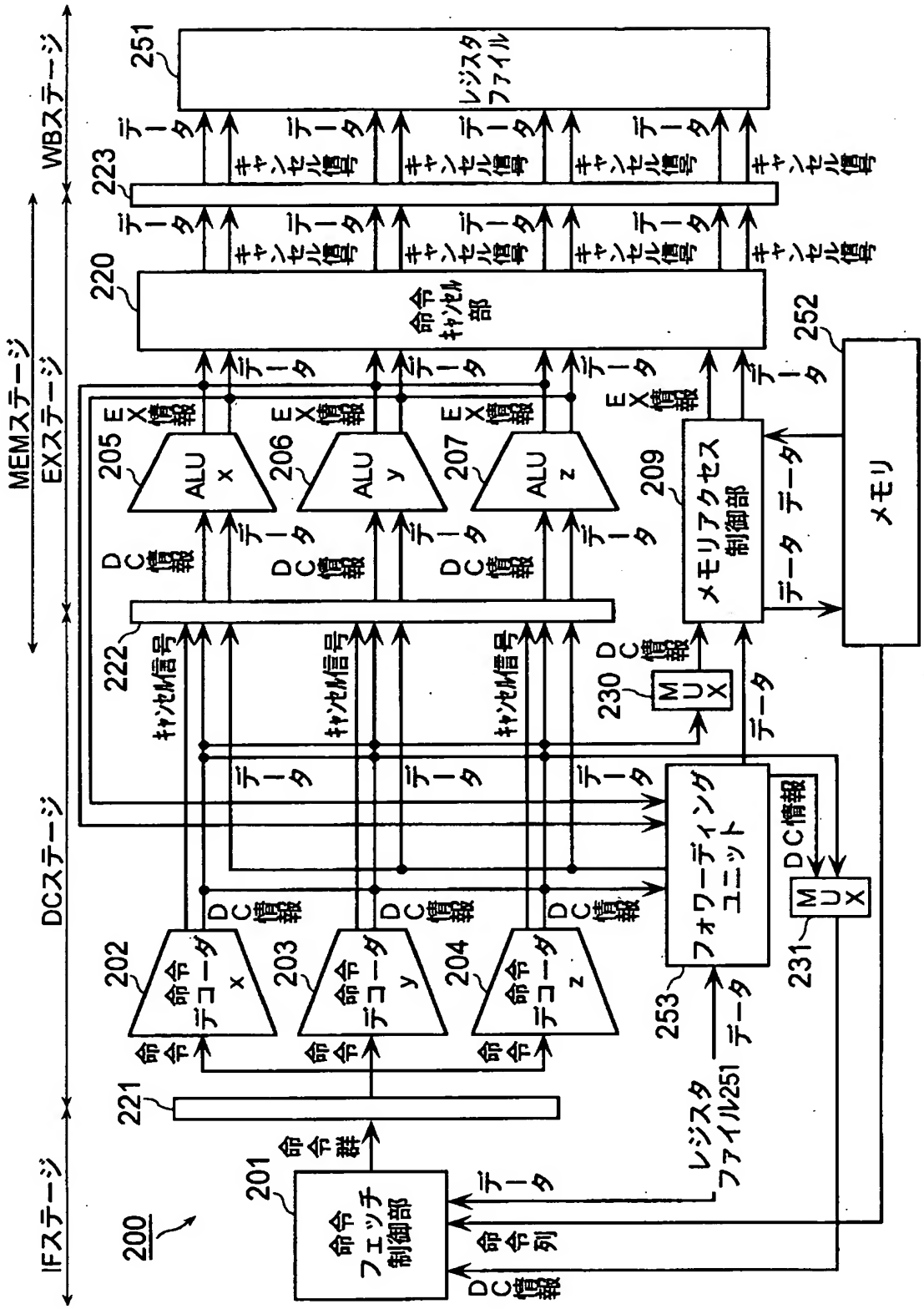
【図２】


> sim
Instruction Step Simulator for VLIW processor Ver.x.xx
Connected with debugger.
Simulating...
W0

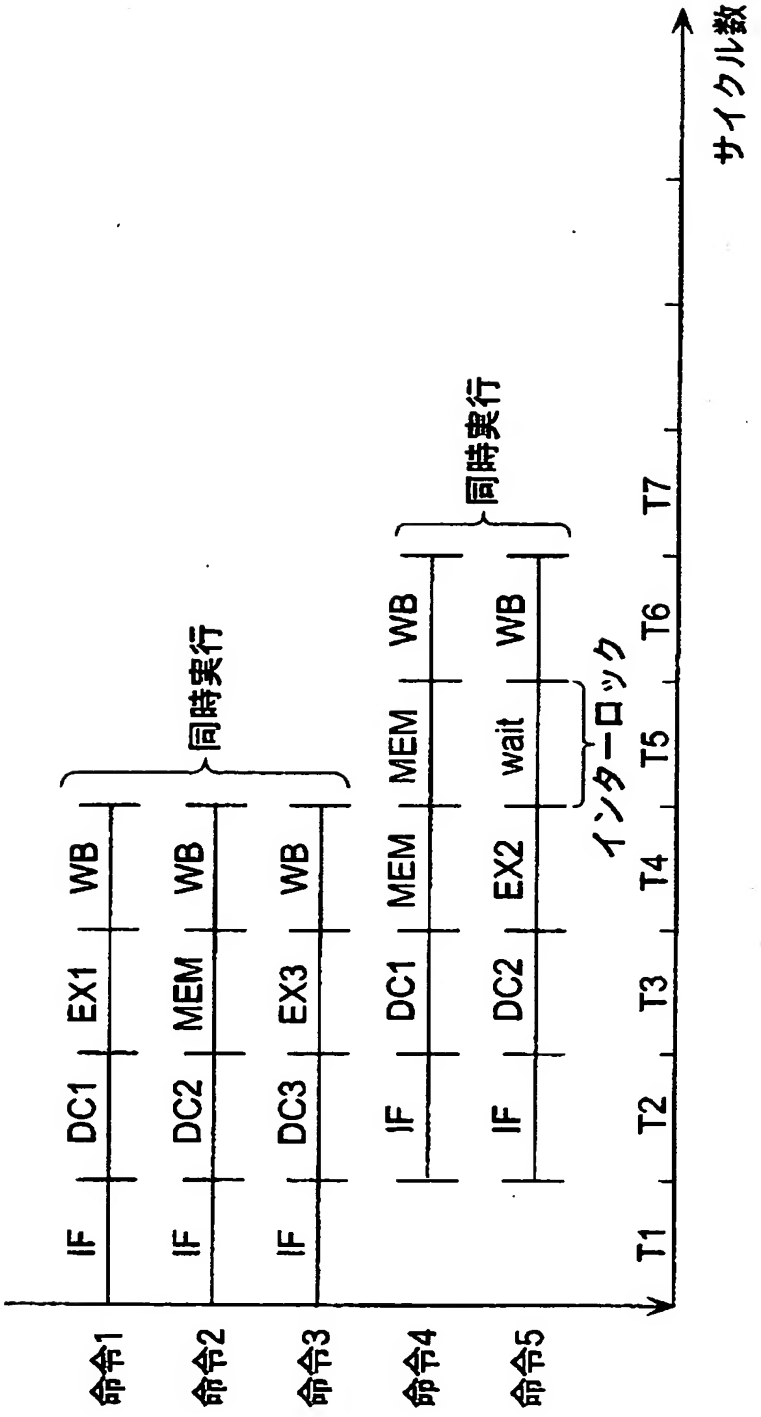
【図３】


W5
PC = 0x00008008
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002000
R3 = 0x00000333
R4 = 0x00000000
R5 = 0x00000000
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00000000
SP = 0x00000ff0
W1
PC LINE FLG MNEMONIC
0x0008000 100 mov R0, 0
0x0008002 101 mov R1, 1 ;;
0x0008004 102 add R0, R1 ;;
0x0008008 103 cmp R0, R3
0x000800A 104 [F0] mov R4, 1
0x000800C 105 ret ;;
103 if(a < b) {
104 d = 1;
105 }
ADDRESS DATA
0x00000000 00, 00, 00, 00, 00, 00, 00, 00
0x00000008 00, 00, 00, 00, 00, 00, 00, 00
0x00000010 00, 00, 00, 00, 00, 00, 00, 00
0x00000018 00, 00, 00, 00, 00, 00, 00, 00
0x00000020 00, 00, 00, 00, 00, 00, 00, 00
0x00000020 00, 00, 00, 00, 00, 00, 00, 00
0x00000030 00, 00, 00, 00, 00, 00, 00, 00
W6
> reset
> s
>
pipline cycles:0029
Instruction Steps:0071
W4
M1
M2
W2
W3

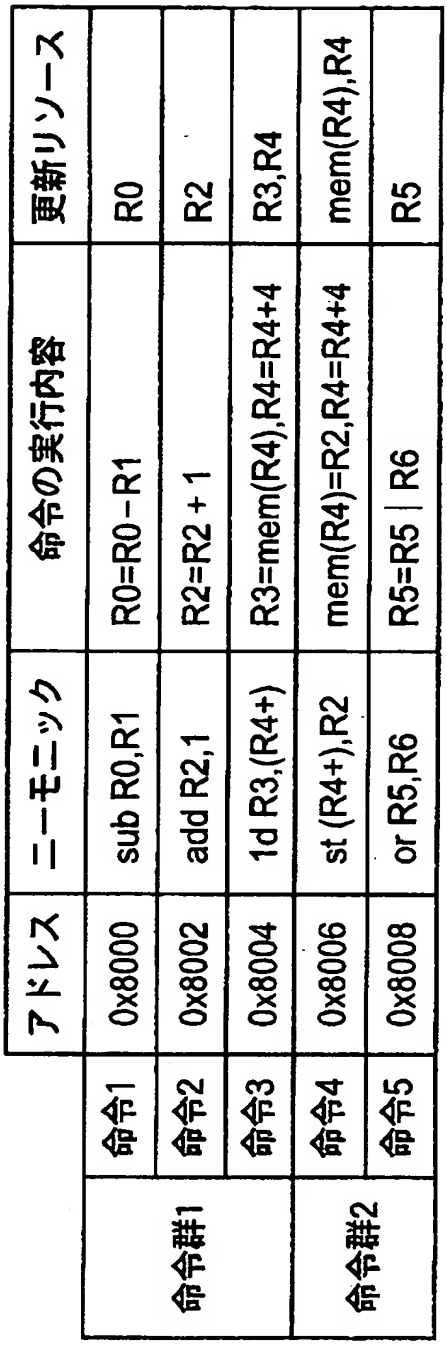
【図４】


WBステージ
MEMステージ
EXステージ
DCステージ
IFステージ
200
201 命令フェッチ制御部
202 命令デコーダx
203 命令デコーダy
204 命令デコーダz
205 ALU x
206 ALU y
207 ALU z
209 メモリアクセス制御部
220 命令キャンセル部
221
222
223
230 MUX
231 MUX
251 レジスタファイル
252 メモリ
253 フォワーディングユニット
データ
DC情報
EX情報
キャンセル信号
命令群
命令列
レジスタファイル251

【図５】


サイクル数
T1
T2
T3
T4
T5
T6
T7
命令1
命令2
命令3
命令4
命令5
IF
DC1
EX1
WB
MEM
EX3
DC2
DC3
EX2
wait
同時実行
インターロック

【図６】

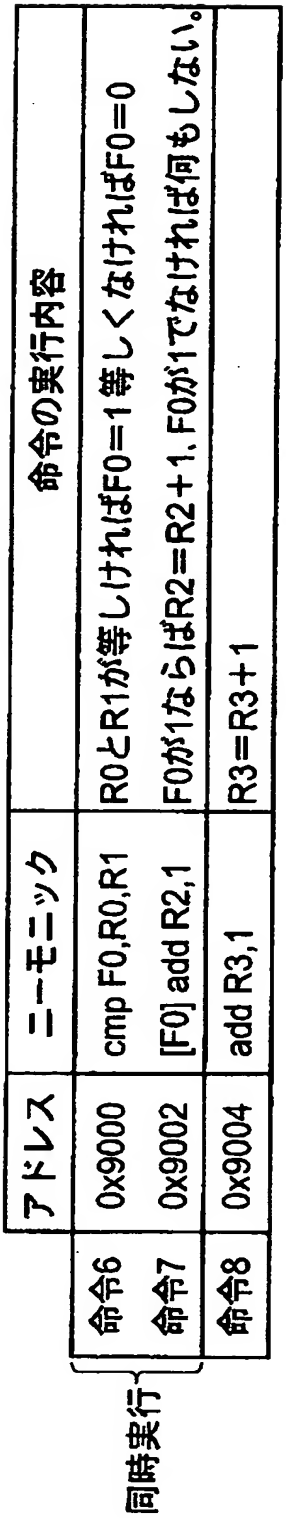


		アドレス	ニーモニック	命令の実行内容	更新リソース
命令群1	命令1	0x8000	sub R0,R1	R0=R0−R1	R0
	命令2	0x8002	add R2,1	R2=R2＋1	R2
	命令3	0x8004	ld R3,(R4+)	R3=mem(R4),R4=R4+4	R3,R4
命令群2	命令4	0x8006	st (R4+),R2	mem(R4)=R2,R4=R4+4	mem(R4),R4
	命令5	0x8008	or R5,R6	R5=R5 | R6	R5

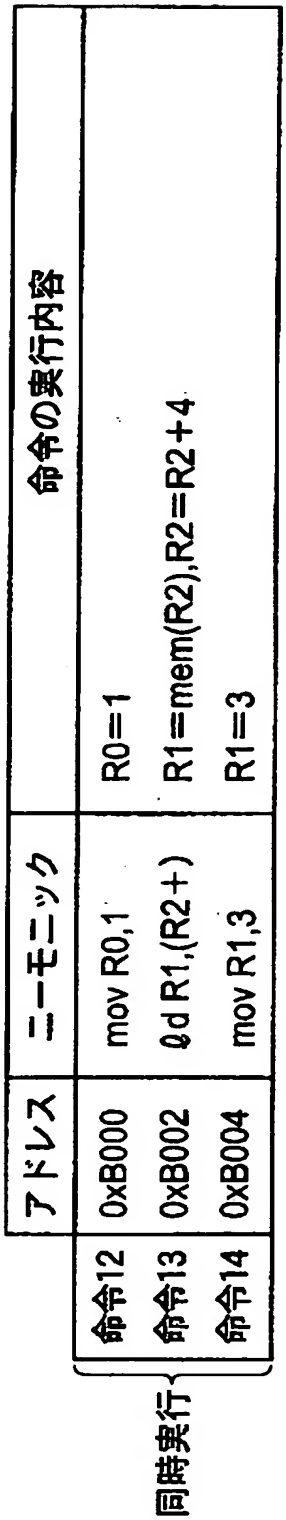
【図７】

サイクル	更新されるリソース
N+1	R0,R2,R3,R4
N+2	＜なし＞
N+3	mem(R4),R4,R5

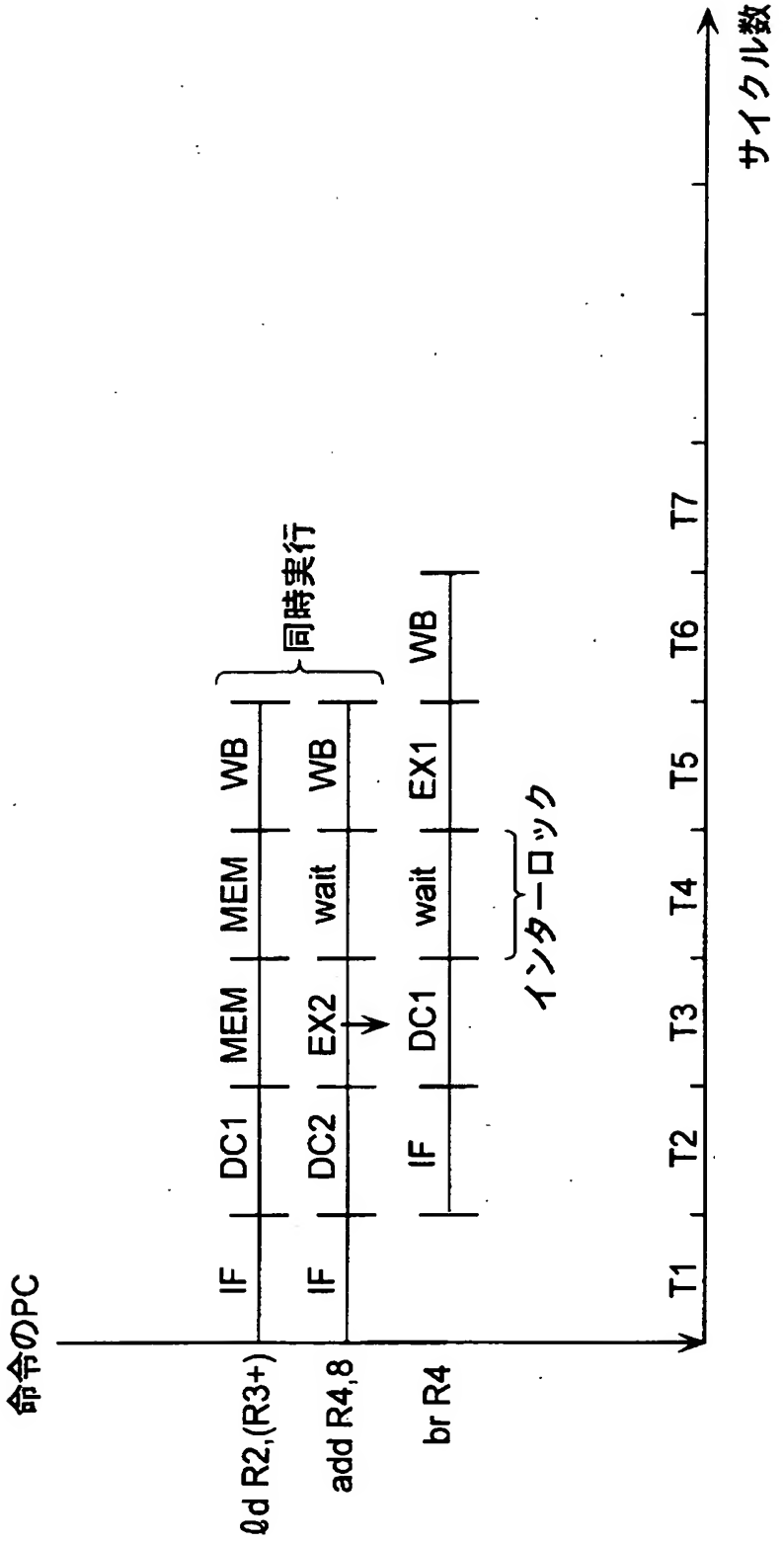
【図８】


同時実行
アドレス
ニーモニック
命令の実行内容
命令6
0x9000
cmp F0,R0,R1
R0とR1が等しければF0=1等しくなければF0=0
命令7
0x9002
[F0] add R2,1
F0が1ならばR2=R2+1, F0が1でなければ何もしない。
命令8
0x9004
add R3,1
R3=R3+1

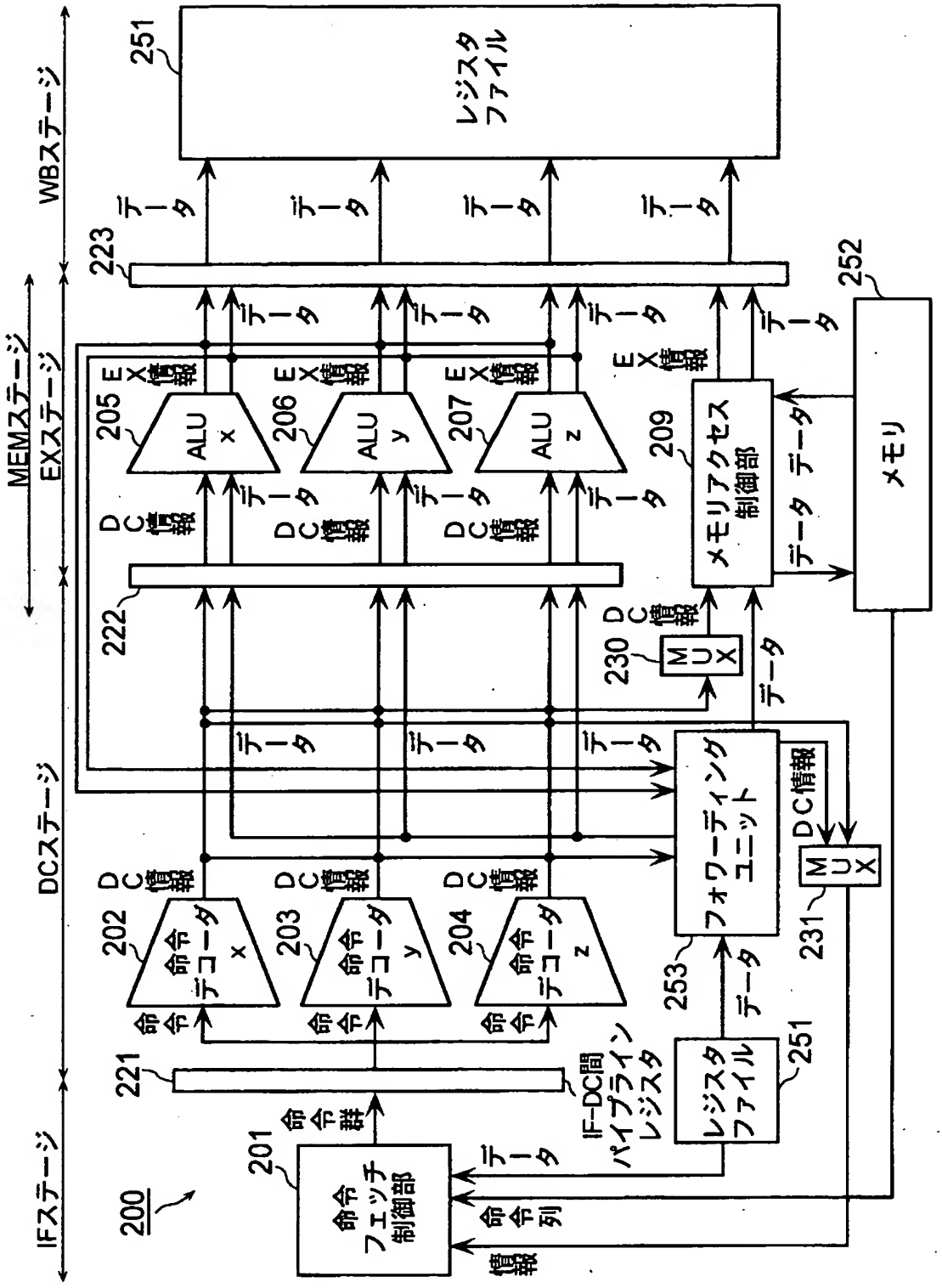
【図９】


同時実行
命令12
命令13
命令14
アドレス
0xB000
0xB002
0xB004
ニーモニック
mov R0,1
ld R1,(R2+)
mov R1,3
命令の実行内容
R0=1
R1=mem(R2),R2=R2+4
R1=3

【図１０】

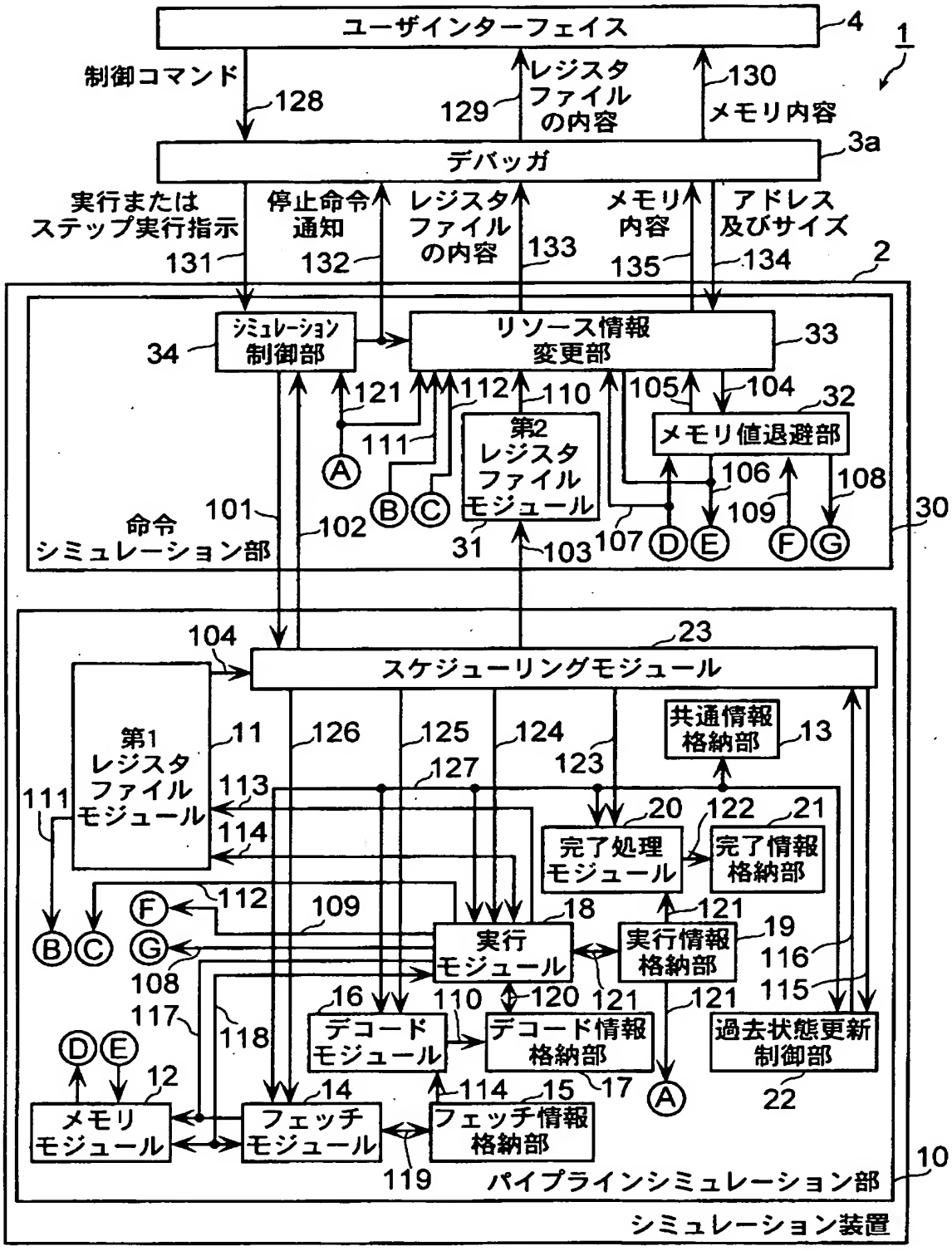

命令のPC
ld R2,(R3+)
add R4,8
br R4
IF
DC1
MEM
WB
IF
DC2
EX2
wait
WB
IF
DC1
wait
EX1
WB
同時実行
インターロック
T1
T2
T3
T4
T5
T6
T7
サイクル数

【図１１】

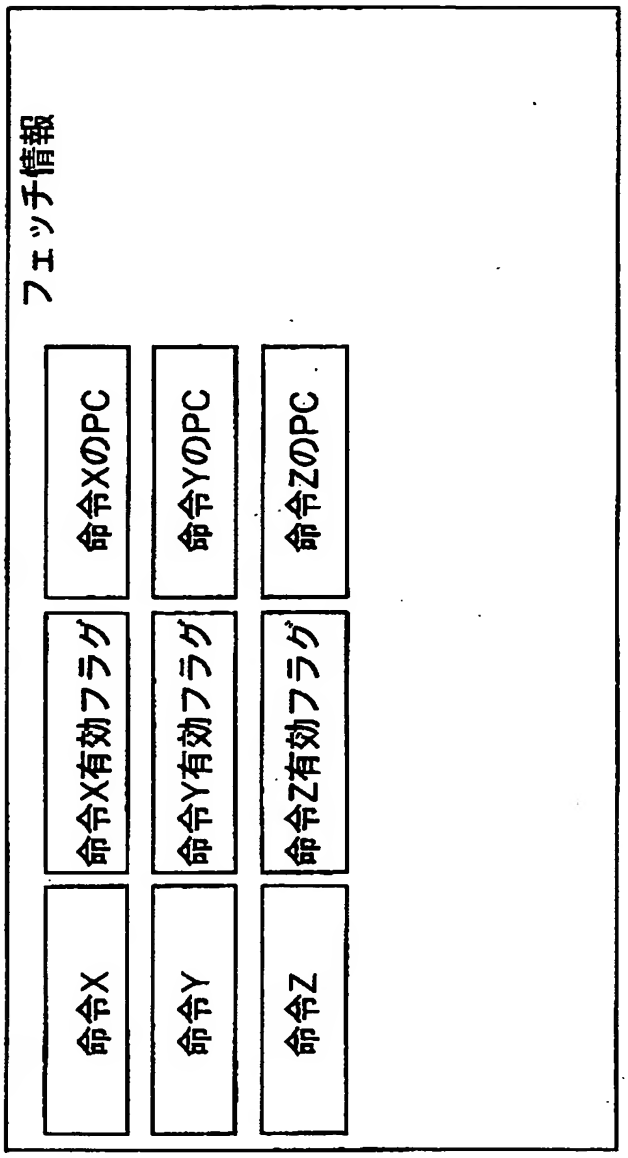


出証特２００３－３０３７４８８

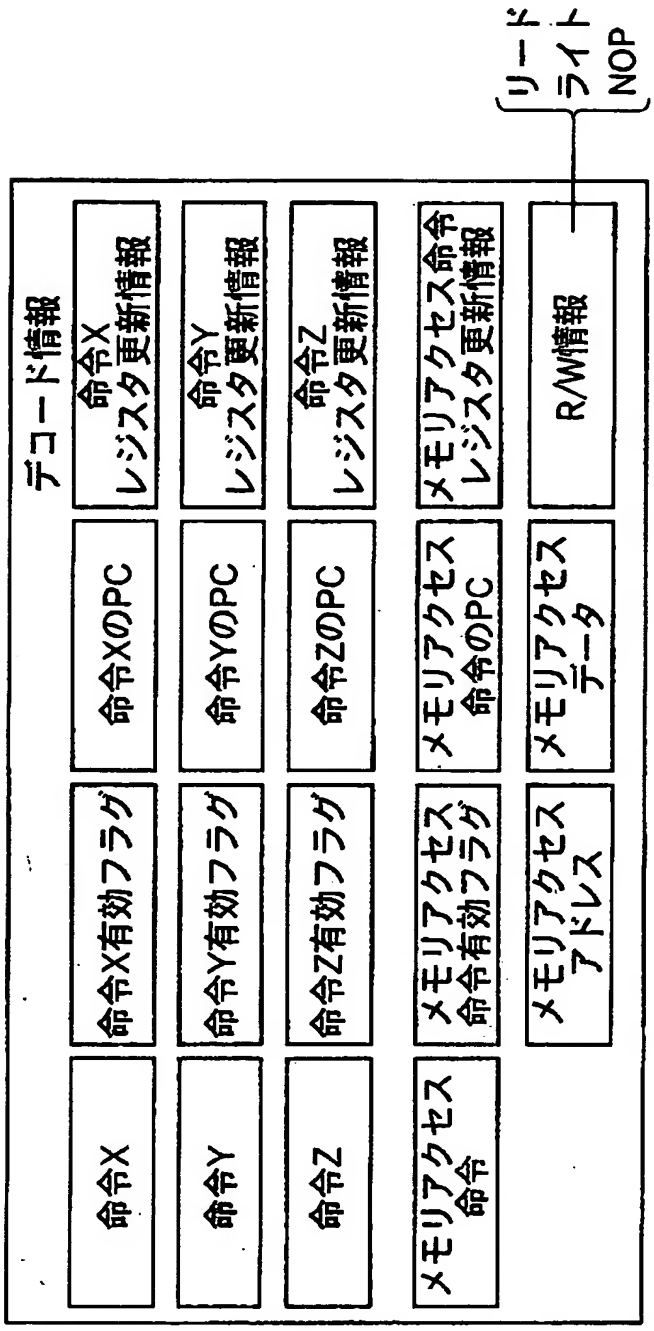
【図１２】


ユーザインターフェイス
4
1
制御コマンド
128
レジスタファイルの内容
129
130
メモリ内容
デバッガ
3a
実行またはステップ実行指示
131
停止命令通知
132
レジスタファイルの内容
133
メモリ内容
135
アドレス及びサイズ
134
2
シミュレーション制御部
34
リソース情報変更部
33
121
112
110
105
104
32
111
第2レジスタファイルモジュール
メモリ値退避部
106
108
101
102
109
命令シミュレーション部
31
103
107
30
104
23
スケジューリングモジュール
第1レジスタファイルモジュール
11
126
125
124
123
共通情報格納部
13
127
113
111
114
20
122
21
完了処理モジュール
完了情報格納部
112
109
18
121
19
実行モジュール
実行情報格納部
116
108
16
110
120
121
121
115
117
118
デコードモジュール
デコード情報格納部
過去状態更新制御部
12
14
114
15
17
22
メモリモジュール
フェッチモジュール
フェッチ情報格納部
119
パイプラインシミュレーション部
10
シミュレーション装置

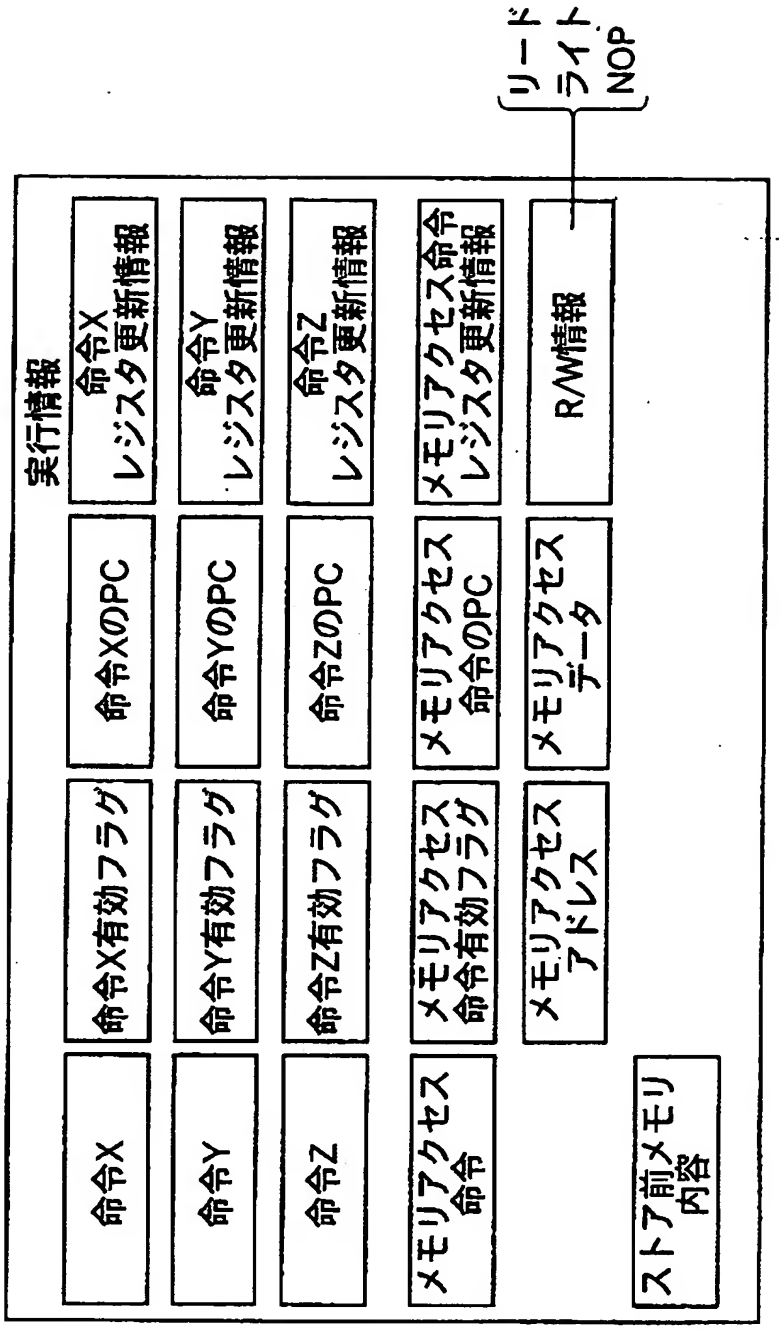
【図１３】


フェッチ情報
命令X
命令X有効フラグ
命令XのPC
命令Y
命令Y有効フラグ
命令YのPC
命令Z
命令Z有効フラグ
命令ZのPC

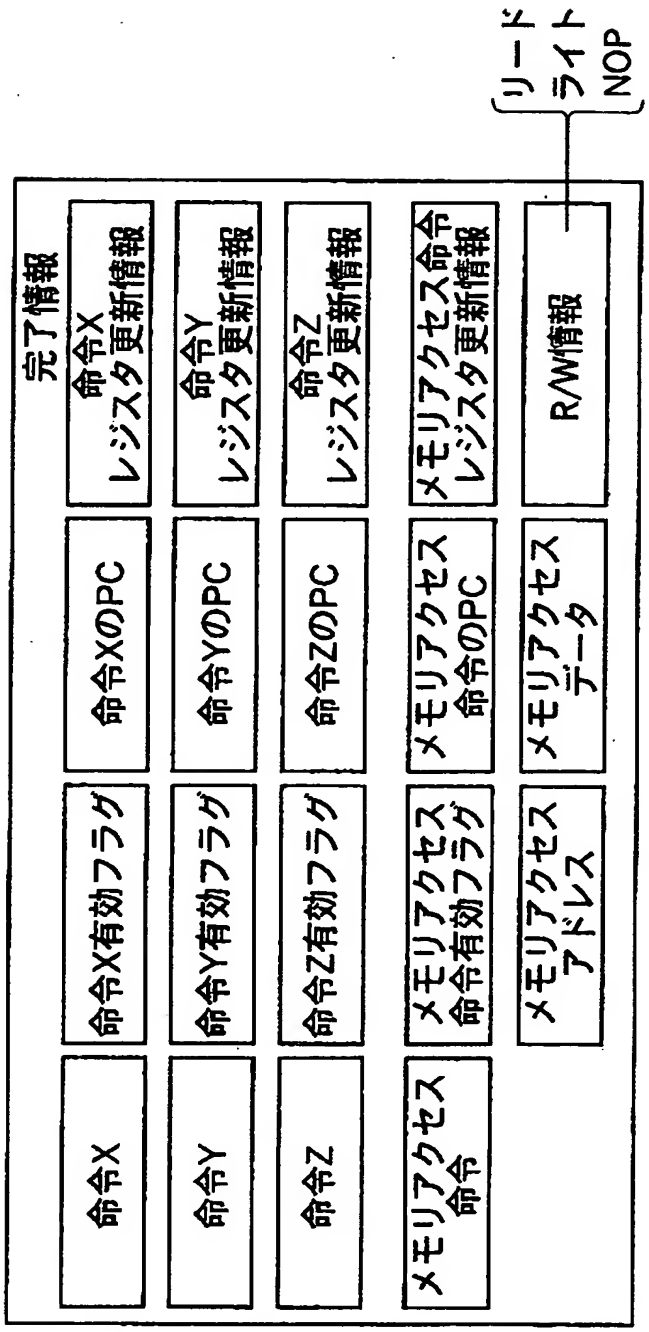
【図１４】


デコード情報
命令X
命令X有効フラグ
命令XのPC
命令X
レジスタ更新情報
命令Y
命令Y有効フラグ
命令YのPC
命令Y
レジスタ更新情報
命令Z
命令Z有効フラグ
命令ZのPC
命令Z
レジスタ更新情報
メモリアクセス
命令
メモリアクセス
命令有効フラグ
メモリアクセス
命令のPC
メモリアクセス命令
レジスタ更新情報
メモリアクセス
アドレス
メモリアクセス
データ
R/W情報
リード
ライト
NOP

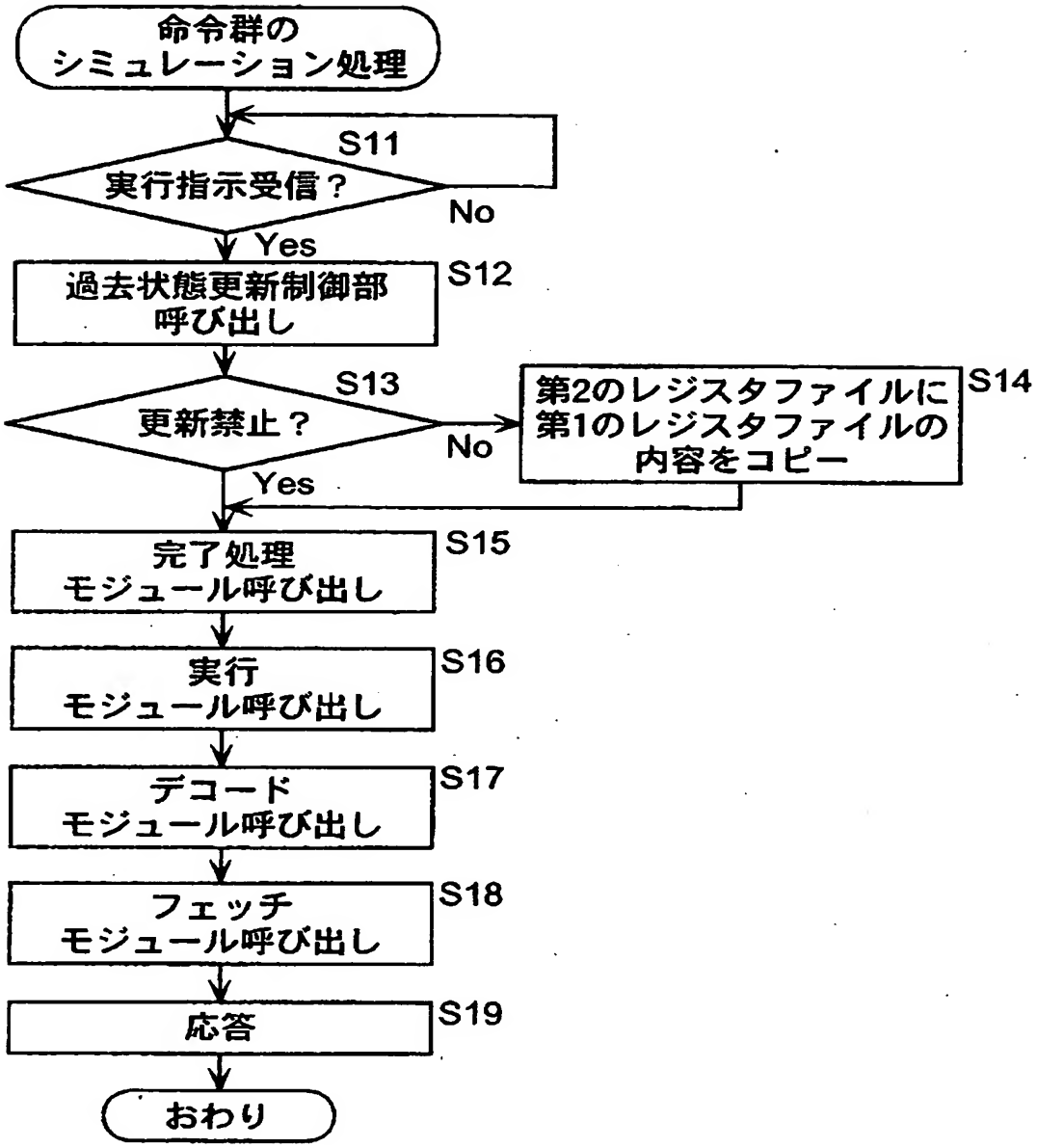
【図１５】


実行情報
命令X
命令X有効フラグ
命令XのPC
命令X
レジスタ更新情報
命令Y
命令Y有効フラグ
命令YのPC
命令Y
レジスタ更新情報
命令Z
命令Z有効フラグ
命令ZのPC
命令Z
レジスタ更新情報
メモリアクセス
命令
メモリアクセス
命令有効フラグ
メモリアクセス
命令のPC
メモリアクセス命令
レジスタ更新情報
メモリアクセス
アドレス
メモリアクセス
データ
R/W情報
ストア前メモリ
内容
リード
ライト
NOP

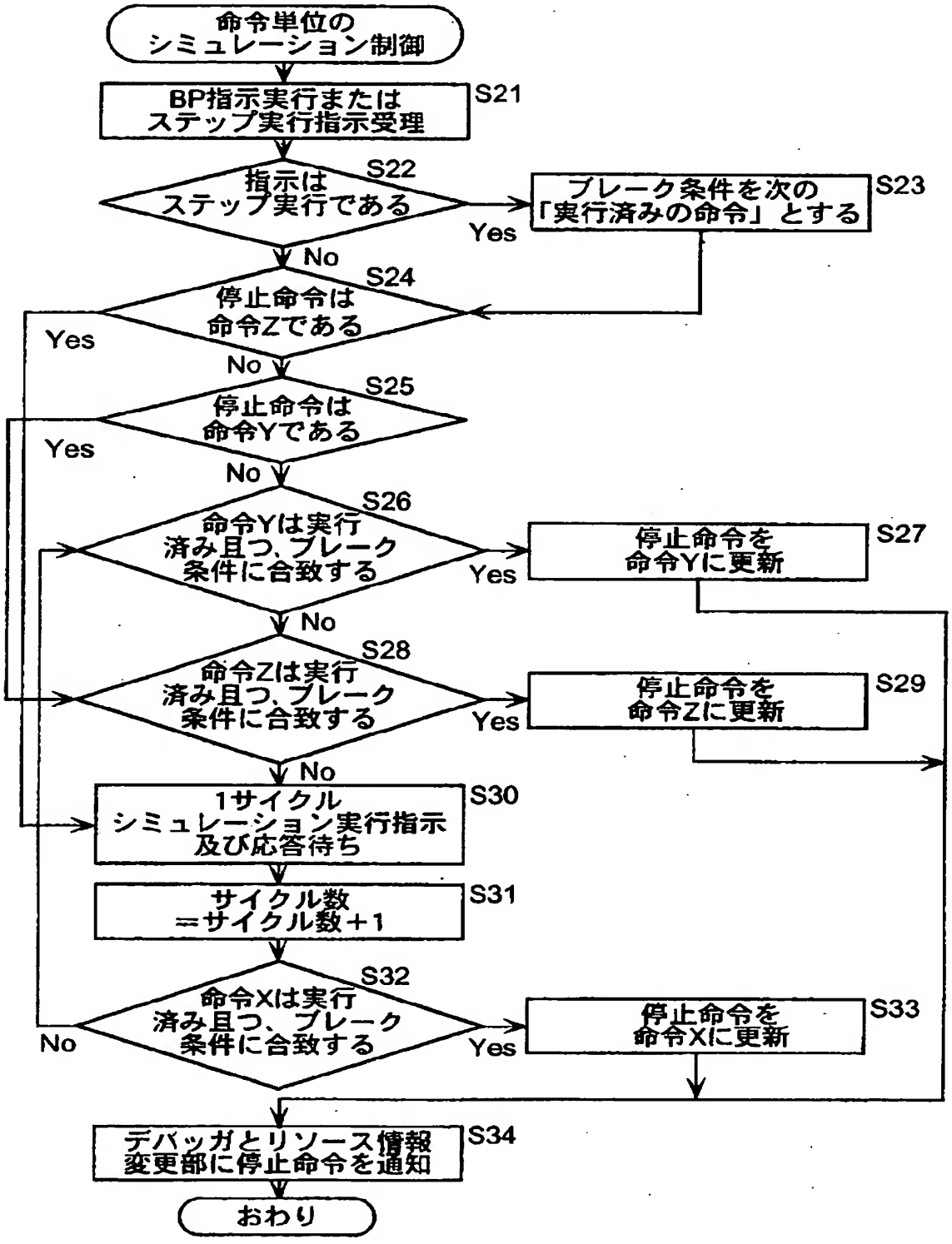
【図１６】


完了情報
命令X
命令X有効フラグ
命令XのPC
命令X
レジスタ更新情報
命令Y
命令Y有効フラグ
命令YのPC
命令Y
レジスタ更新情報
命令Z
命令Z有効フラグ
命令ZのPC
命令Z
レジスタ更新情報
メモリアクセス
命令
メモリアクセス
命令有効フラグ
メモリアクセス
命令のPC
メモリアクセス命令
レジスタ更新情報
メモリアクセス
アドレス
メモリアクセス
データ
R/W情報
リード
ライト
NOP

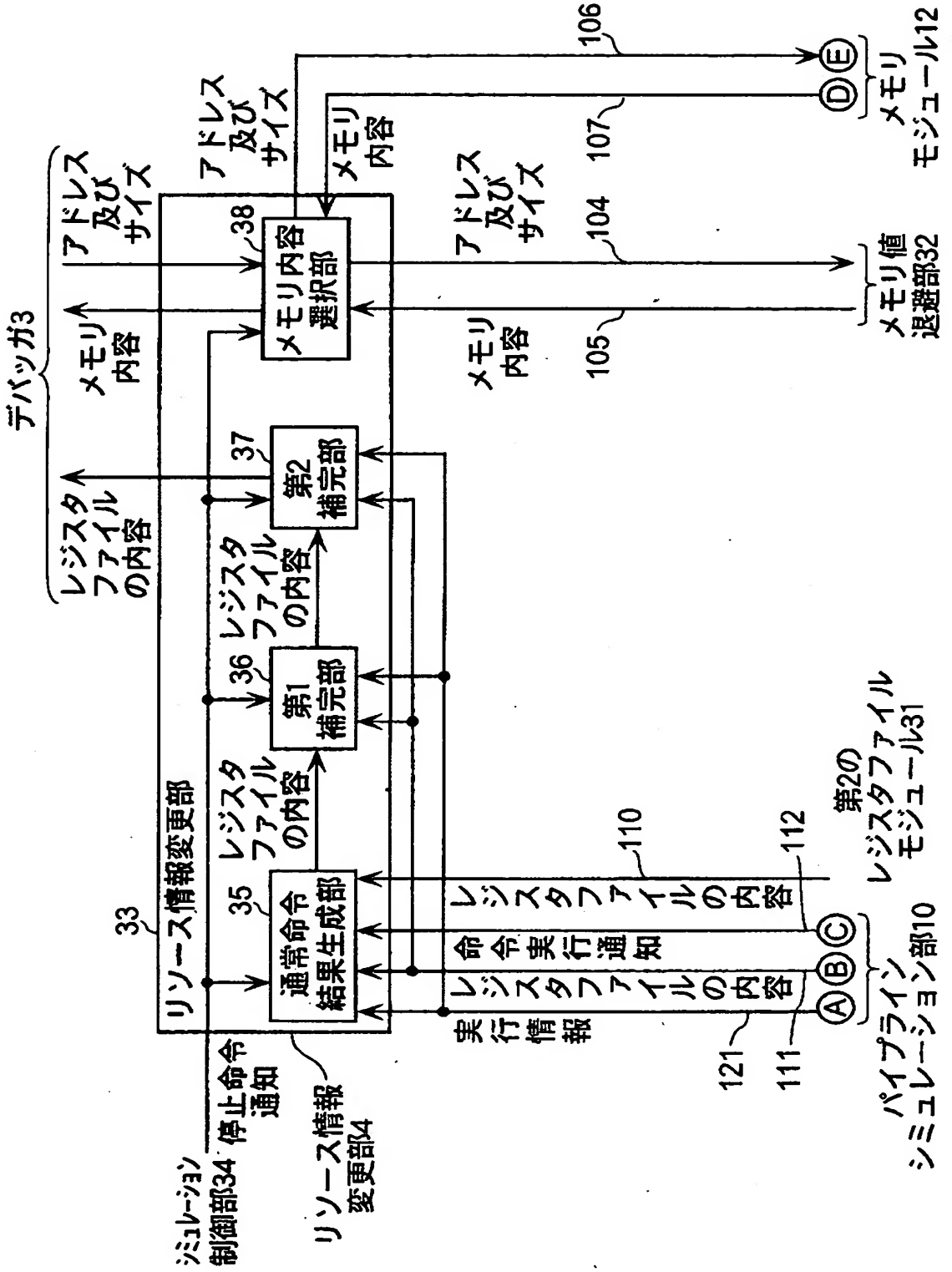
【図１７】


命令群の
シミュレーション処理
S11
実行指示受信？
No
Yes
過去状態更新制御部
呼び出し
S12
S13
更新禁止？
No
第2のレジスタファイルに
第1のレジスタファイルの
内容をコピー
S14
Yes
完了処理
モジュール呼び出し
S15
実行
モジュール呼び出し
S16
デコード
モジュール呼び出し
S17
フェッチ
モジュール呼び出し
S18
応答
S19
おわり

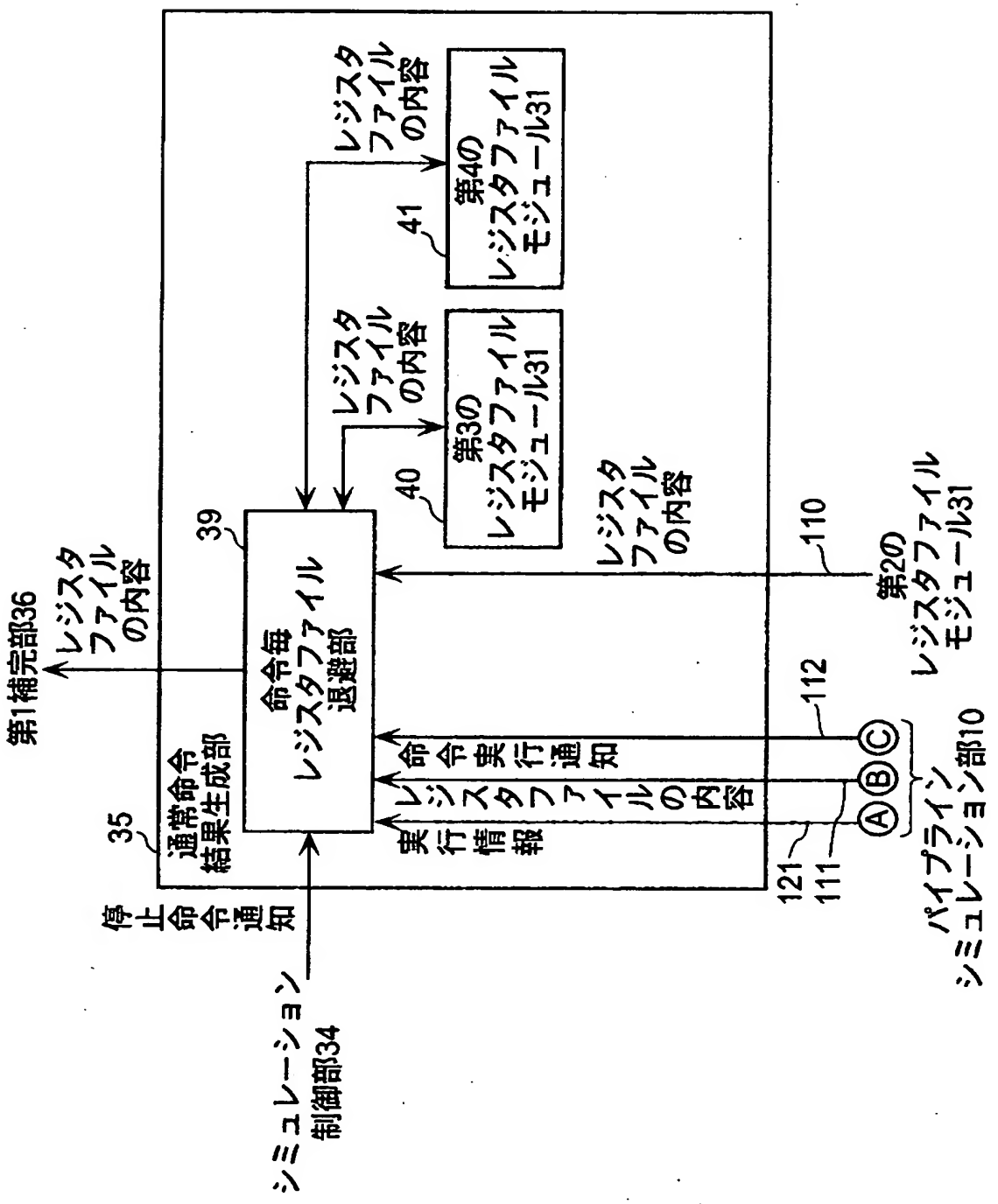
【図１８】


命令単位の
シミュレーション制御
BP指示実行または
ステップ実行指示受理
S21
指示は
ステップ実行である
S22
Yes
ブレーク条件を次の
「実行済みの命令」とする
S23
No
停止命令は
命令Zである
S24
Yes
No
停止命令は
命令Yである
S25
Yes
No
命令Yは実行
済み且つ、ブレーク
条件に合致する
S26
Yes
停止命令を
命令Yに更新
S27
No
命令Zは実行
済み且つ、ブレーク
条件に合致する
S28
Yes
停止命令を
命令Zに更新
S29
No
1サイクル
シミュレーション実行指示
及び応答待ち
S30
サイクル数
＝サイクル数＋1
S31
命令Xは実行
済み且つ、ブレーク
条件に合致する
S32
No
Yes
停止命令を
命令Xに更新
S33
デバッガとリソース情報
変更部に停止命令を通知
S34
おわり

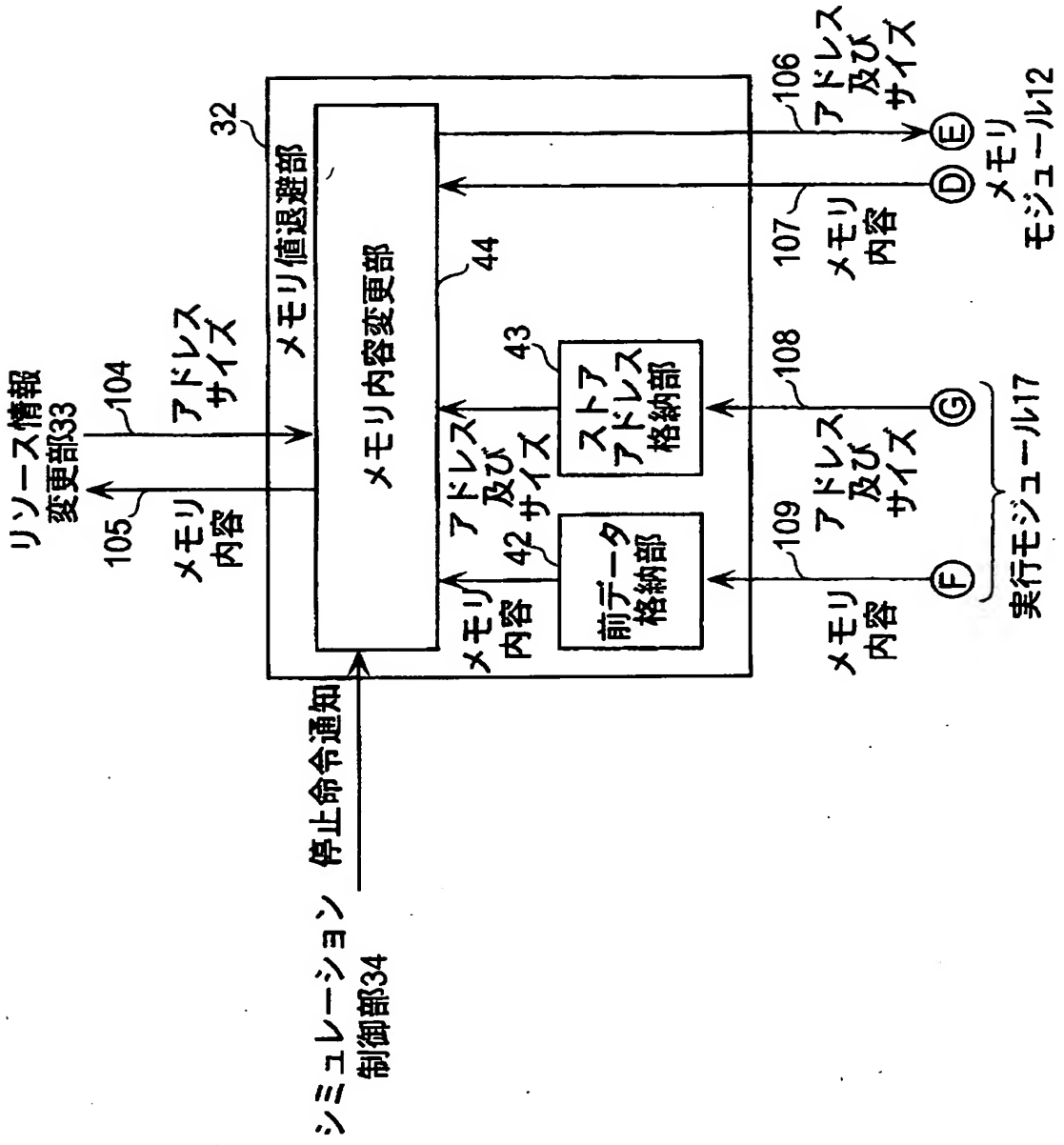
【図１９】


デバッガ3
アドレス及びサイズ
メモリ内容
レジスタファイルの内容
メモリ内容選択部
38
第2補完部
37
第1補完部
36
通常命令結果生成部
35
リソース情報変更部
33
シミュレーション制御部34
停止命令通知
リソース情報変更部4
レジスタファイルの内容
命令実行通知
実行情報
110
112
121
111
第2のレジスタファイルモジュール31
パイプラインシミュレーション部10
Ⓐ Ⓑ Ⓒ
Ⓓ Ⓔ
メモリモジュール12
106
107
104
105
メモリ値退避部32

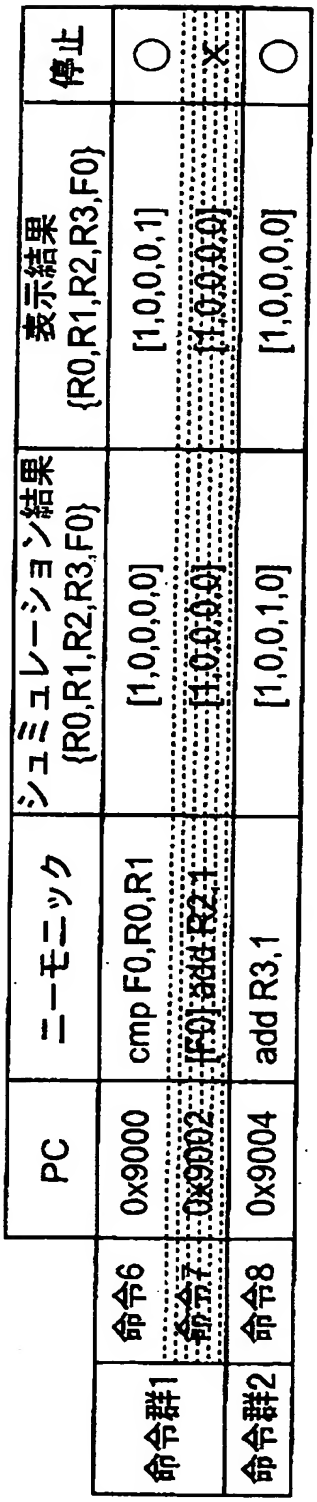
【図２０】


第1補完部36
レジスタファイルの内容
39
命令毎レジスタファイル退避部
35
通常命令結果生成部
停止命令通知
シミュレーション制御部34
レジスタファイルの内容
第4のレジスタファイルモジュール31
41
レジスタファイルの内容
第3のレジスタファイルモジュール31
40
レジスタファイルの内容
110
第2のレジスタファイルモジュール31
112
命令実行通知
レジスタファイルの内容
実行情報
Ⓐ Ⓑ Ⓒ
121
111
パイプラインシミュレーション部10

【図２１】


リソース情報
変更部33
104
アドレス
サイズ
105
メモリ
内容
32
メモリ値退避部
メモリ内容変更部
44
43
ストア
アドレス
格納部
アドレス
及び
サイズ
42
前データ
格納部
メモリ
内容
106
アドレス
及び
サイズ
Ⓔ
Ⓓ
メモリ
モジュール12
107
メモリ
内容
108
アドレス
及び
サイズ
Ⓖ
109
アドレス
及び
サイズ
Ⓕ
メモリ
内容
実行モジュール17
シミュレーション
制御部34
停止命令通知

【図２２】

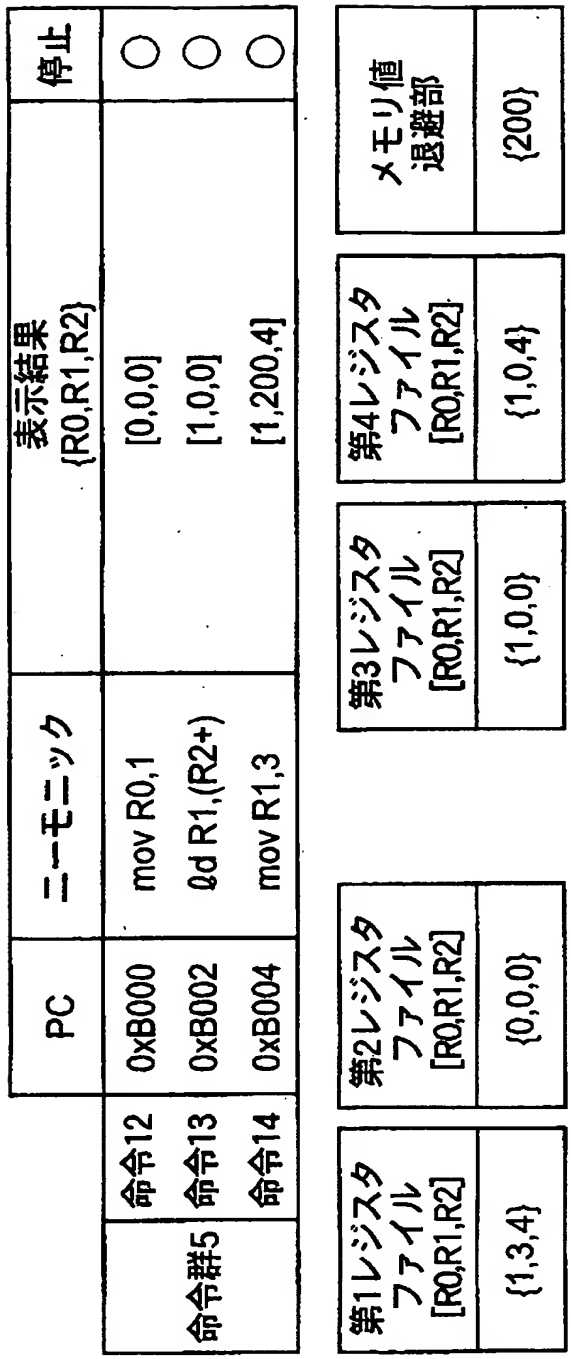


		PC	ニーモニック	シュミュレーション結果 {R0,R1,R2,R3,F0}	表示結果 {R0,R1,R2,R3,F0}	停止
命令群1	命令6	0x9000	cmp F0,R0,R1	[1,0,0,0,0]	[1,0,0,0,1]	○
	命令7	0x9002	[F0] add R2,1	[1,0,1,0,0]	[1,0,1,0,0]	X
命令群2	命令8	0x9004	add R3,1	[1,0,1,1,0]	[1,0,1,0,0]	○

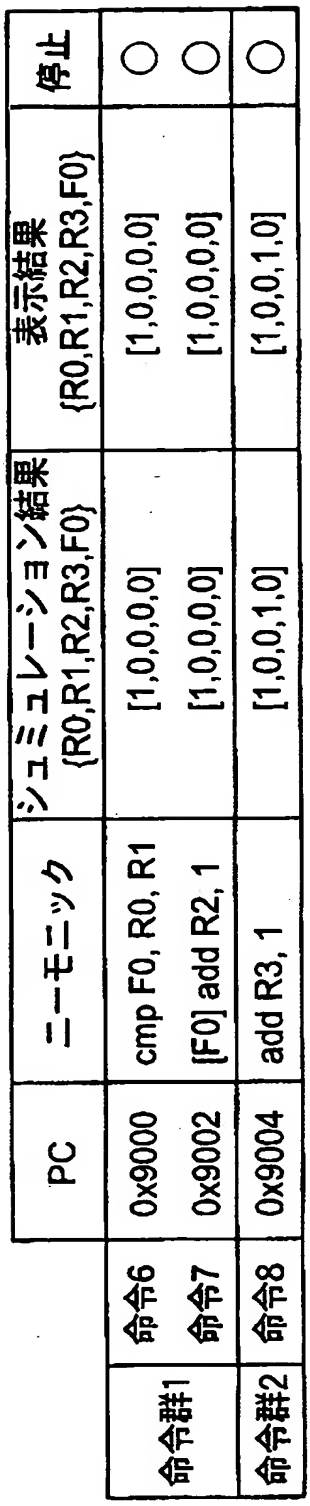
【図２３】

		PC	ニーモニック	表示結果 {R0,R1,R2,R3,R4,R5,R6}	停止
命令群1	命令1	0x8000	sub_R0,R1	[10,5,0,0,0,1,2]	○
	命令2	0x8002	add R2,1	[5,5,0,0,0,1,2]	○
	命令3	0x8004	ld R3,(R4+)	[5,5,1,0,0,1,2]	○
命令群2	命令4	0x8006	st (R4+),R2	[5,5,1,100,4,1,2]	○
	命令5	0x8008	or R5,R6	[5,5,1,100,8,1,2]	○

【図２４】


PC
ニーモニック
表示結果
{R0,R1,R2}
停止
命令群5
命令12
命令13
命令14
0xB000
0xB002
0xB004
mov R0,1
ld R1,(R2+)
mov R1,3
[0,0,0]
[1,0,0]
[1,200,4]
○
○
○
第1レジスタ
ファイル
[R0,R1,R2]
{1,3,4}
第2レジスタ
ファイル
[R0,R1,R2]
{0,0,0}
第3レジスタ
ファイル
[R0,R1,R2]
{1,0,0}
第4レジスタ
ファイル
[R0,R1,R2]
{1,0,4}
メモリ値
退避部
{200}

【図２５】



		PC	ニーモニック	シュミュレーション結果 {R0,R1,R2,R3,F0}	表示結果 {R0,R1,R2,R3,F0}	停止
命令群1	命令6	0x9000	cmp F0, R0, R1	[1,0,0,0,0]	[1,0,0,0,0]	○
	命令7	0x9002	[F0] add R2, 1	[1,0,0,0,0]	[1,0,0,0,0]	○
命令群2	命令8	0x9004	add R3, 1	[1,0,0,1,0]	[1,0,0,1,0]	○

【書類名】　　要約書

【要約】

【課題】　複数命令を同時実行するプロセッサを対象とし、命令単位の実行過程がわかるようにシミュレーションを行うシミュレーション装置を提供する。

【解決手段】　シミュレーション装置２においてパイプラインシミュレーション部１０は、同時実行されるべき複数命令からなる命令群をシミュレーションし、命令シミュレーション部３０は、パイプラインシミュレーション部１０によるシミュレーション結果に基づいて、前記命令群中の命令毎のシミュレーション結果を生成する。

【選択図】図１.2

認定・付加情報

特許出願の番号	特願２００２－３６０３６２
受付番号	５０２０１８８１４２６
書類名	特許願
担当官	第七担当上席　００９６
作成日	平成１４年１２月１３日

＜認定情報・付加情報＞

【提出日】　平成14年12月12日

次頁無

出証特２００３－３０３７４８８

出　願　人　履　歴　情　報

識別番号　　〔０００００５８２１〕

１．変更年月日　　１９９０年　８月２８日
［変更理由］　新規登録
住　所　　大阪府門真市大字門真１００６番地
氏　名　　松下電器産業株式会社